Introduction

- Firewall mentality
  - on the Internet the cornerstone of security is the notion of a firewall
  - a logically bounded system within a physically unbounded one
  - bounded-system thinking within unbounded domains can lead to security designs and architectures that are fundamentally flawed from a survivability perspective
  - firewalls are state of the art for security systems but not for survivable systems
  - firewalls are passive and implement filter functions
  - how “managed” is the firewall, i.e. is it configured effectively?
    » Danger: False sense of security!

Introduction

- Defining Requirements for Survivable Systems
  - Survivability requirements depend on main issues:
    » system scope
    » criticality
    » consequences of failure and interruption of service
  - In addition to software functionality, survivability must address requirements for
    » software usage,
    » development,
    » operation,
    » evolution.
Introduction

- Defining Requirements for Survivable Systems (cont)
  - New paradigm characterized by:
    » distributed services
    » distributed logic
    » distributed code
    » distributed hardware
    » shared communications and routing infrastructure
    » diminished trust
    » lack of unified administrative control

  - Paradigms formidable effort for software engineering research
    » traditional computer security measures are augmented by comprehensive system survivability strategies

Introduction

- Survivability Requirements
  - we now discuss each of the following topics briefly:
    » System/Survivability Requirements
    » Usage/Intrusion Requirements
    » Development Requirements
    » Operations Requirements
    » Evolution Requirements
Survivability Requirements
- refer to capability of system to
  » deliver essential services in the presence of intrusions
  » recover full services

- system should be organized into essential and non-essential services
  » essential services
    ■ must be maintained even during successful intrusion
    ■ may have different levels,
      - "prioritize by severity and duration of intrusion"
    ■ must be augmented with survivability requirements
  » non-essential services
    ■ are recovered after intrusion has been handled
  » in this paper the view is "binary", however that must not be the case
Introduction

◆ System Requirements
  – describe traditional user functions a system must provide
  – example: network management system must provide
    » monitoring of system, performance adjustments, etc.
  – may include non-functional aspects
    » timing, performance, reliability

Introduction

◆ Survivability Requirements (cont.)
  – COTS components not developed with survivability objective
    » may provide both essential and non-essential services
    » may require functional requirements for isolation and control (using wrappers and filters)
  – survivability imposes new requirements on system
    » resistance to, recognition of and recovery from malicious acts
    » adaptation and evolution
Introduction

System/Survivability Requirements (cont.)
- term *emergent behavior requirements* at network level
  - underlines that requirements are not associated with a particular node, but emerge from the collective behavior
  - issue is survivability of the overall network capability
    - e.g., message routing in the presence of topology degradation
- capability of adapting
  - behavior, function, resource allocation
  - resources may be shifted from non-essential to essential services
System/Survivability Requirements (cont.)

- survivability requirements may vary greatly
  - small systems may only have non-essential services
    - (recovery in hours)
  - large systems (large networks) may have core set of essential services, automated intrusion detection
    - (recovery in minutes)
  - embedded control systems may require essential services in real-time
    - (recovery in milliseconds)

- no free lunch
  - attainment and maintenance of survivability consumes resources in
    - development,
    - operation,
    - evolution
  - cost and risk analysis to manage resources wisely

Usage/Intrusion Requirements

- testing must demonstrate
  - correct performance of essential and non-essential system services
  - survivability of essential services under intrusion
  - “How does one do this?”

- but this depends totally on the system’s use
  - use of usage scenarios derived from usage models

- usage models
  - are developed from usage requirements
  - they specify usage environments and scenarios of system use

- usage requirements for essential and non-essential services must be defined in parallel with system and survivability requirements
Introduction

- Usage/Intrusion Requirements (cont.)
  - relationship between legitimate and intrusion use
  - intruder may engage in scenarios beyond legitimate scenarios
    » but may use legitimate usage

- Development Requirements
  - ***stringent requirements*** on system development and testing practices
  - inadequate functionality and software errors can have devastating effects (provide opportunities for intrusion)
  - ***sound engineering practices*** are required
  - this also holds for legacy and COTS software components
Introduction

Development Requirements (cont.)
- Some example requirements for survivable-system development and testing practices:
  - For some you will say: Yeah right! - How big is the system?!
    @#$%
    » Precisely specify the system’s required functions in all possible circumstances of system use.
    » Verify the correctness of system implementations with respect to the functional specifications.
    » Specify function usage in all possible circumstances of system use, including intruder use.
    » Test and certify the system based on function usage and statistical methods.
    » Establish permanent readiness teams for system monitoring,

Introduction

Operation Requirements
- demands for system operation and administration
- defining and communicating survivability policies
- monitoring system use
- response to intrusion
- evolving system functions as needed to ensure survivability under consideration of changes over time in usage environments and intrusion patterns
Introduction

- Evolution Requirements
  - system evolution responds to user requirements for new functions
  - evolution necessary to respond to increasing intruder knowledge of system behavior and structure
  - survivability requires system capabilities to evolve faster than intrusion knowledge
  - rapid evolution prevents intruders from accumulating information about otherwise invariant system behavior

Requirements Definition for Essential Services

- set of essential services must form viable subsystem
  » complete and coherent

- What if multiple levels of essential services are used?
  » each level must be examined for completeness and coherence
  » requirements needed to define transition to and from different levels

- Provisions for tracing survivability requirements through design, code and test must be established
Introduction

- Requirements Definition for Survivable Services
  - need to define a set of requirements for survivable services
  - four general categories
    » resistance
    » recognition
    » recovery
    » adaptation & evolution

  - these requirements operate in environment with phases of intrusions
    » penetration
    » exploration
    » exploitation

- Penetration Phase
  - attempt to gain access through various attack scenarios
  - amateur and professional hackers
  - capitalization on known system vulnerabilities

- Exploration Phase
  - system has been penetrated
  - intruder is exploring internal system organization
  - learns to exploit the access to achieve intrusion objective

- Exploitation Phase
  - performance of operations to compromise system capabilities
Introduction

- Requirements Definition for Survivable Services (cont.)

  - Exploitation Phase (cont.)
    
    » penetration at user level as stepping stone to find root-level vulnerabilities
    
    » exploit those vulnerabilities to achieve root-level penetration
    
    » compromise of the weakest host in network as stepping stone to compromise more protected hosts.

Survivability Life Cycle Definition

- This discussion is based on section 3 of:

  Survivable Network Analysis Method

  CMU/SEI-2000-TR-013
  ESC-TR-2000-013

  Nancy R. Mead
  Robert J. Ellison
  Richard C. Linger
  Thomas Longstaff
  John McHugh
**Waterfall Model**

- Model utilizes following steps:
  - Document of system concept
  - Identify of system requirements and analyze them
  - Break the System into Pieces
    - Architectural Design
  - Design Each Piece
    - Detailed Design
  - Code the System Components and Test them Individually
    - Coding, Debugging, and Unit Testing
  - Integrate the Pieces and Test the System
    - System Testing
  - Deploy the System and Operate it

**Waterfall vs. Spiral Model**

- Shortcomings of Waterfall Model
  - does not quite apply to today’s development realities
    - represents a “linear process in batch-oriented world”
  - missing
    - flexibility
    - robustness
    - risk management capabilities
- Spiral Model
  - “accommodates” activities such as prototyping, reuse, automatic coding as part of the process”
  - very important to us is risk management
  - need to augment spiral model by survivability considerations
**Spiral Model**

- Overcome limitations of Waterfall Model

![Image of Spiral Model diagram]

*Figure 1: A Project Spiral Cycle [source CMU/SEI-2000-TR-013]*

**Specialization of Spiral Model**

- **Primary Motivation/Driver:** Functionality, Performance, Dependability, Scalability, ...  
- **Primary Constraints:** Mission, Organization, Environment, Cost, Schedule, Knowledge, Technology, ...

![Image of Specialization diagram]

*Figure 2: Specialization of the Spiral Model for Survivability Driver* [source CMU/SEI-2000-TR-013]
## Life-Cycle Activities

Table 2: Life-Cycle Activities and Corresponding Survivability Elements

<table>
<thead>
<tr>
<th>Life-Cycle Activities</th>
<th>Key Survivability Elements</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission Definition</td>
<td>Analysis of mission criticality and consequences of failure</td>
<td>Estimation of cost impact of denial of service attacks</td>
</tr>
<tr>
<td>Concept of Operations</td>
<td>Definition of system capabilities in adverse environments</td>
<td>Enumeration of critical mission functions that must withstand attacks</td>
</tr>
<tr>
<td>Project Planning</td>
<td>Integration of survivability into life-cycle activities</td>
<td>Identification of defensive coding techniques for implementation</td>
</tr>
<tr>
<td>Requirements Definition</td>
<td>Definition of survivability requirements from mission perspective</td>
<td>Definition of access requirements for critical system assets during attacks</td>
</tr>
<tr>
<td>System Specification</td>
<td>Specification of essential service and intrusion scenarios</td>
<td>Definition of steps that compose critical system transactions</td>
</tr>
<tr>
<td>System Architecture</td>
<td>Integration of survivability strategies into architecture definition</td>
<td>Creation of network facilities for replication of critical data assets</td>
</tr>
<tr>
<td>System Design</td>
<td>Development and verification of survivability strategies</td>
<td>Correctness verification of data encryption algorithms</td>
</tr>
<tr>
<td>System Implementation</td>
<td>Application of survivability coding and implementation techniques</td>
<td>Definition of methods to avoid buffer overflow vulnerabilities</td>
</tr>
<tr>
<td>System Testing</td>
<td>Treatment of intruders as users in testing and certification</td>
<td>Addition of intrusion usage to usage models for statistical testing</td>
</tr>
<tr>
<td>System Evolution</td>
<td>Improvement of survivability to prevent degradation over time</td>
<td>Redefinition of architecture in response to changing threat environment</td>
</tr>
</tbody>
</table>