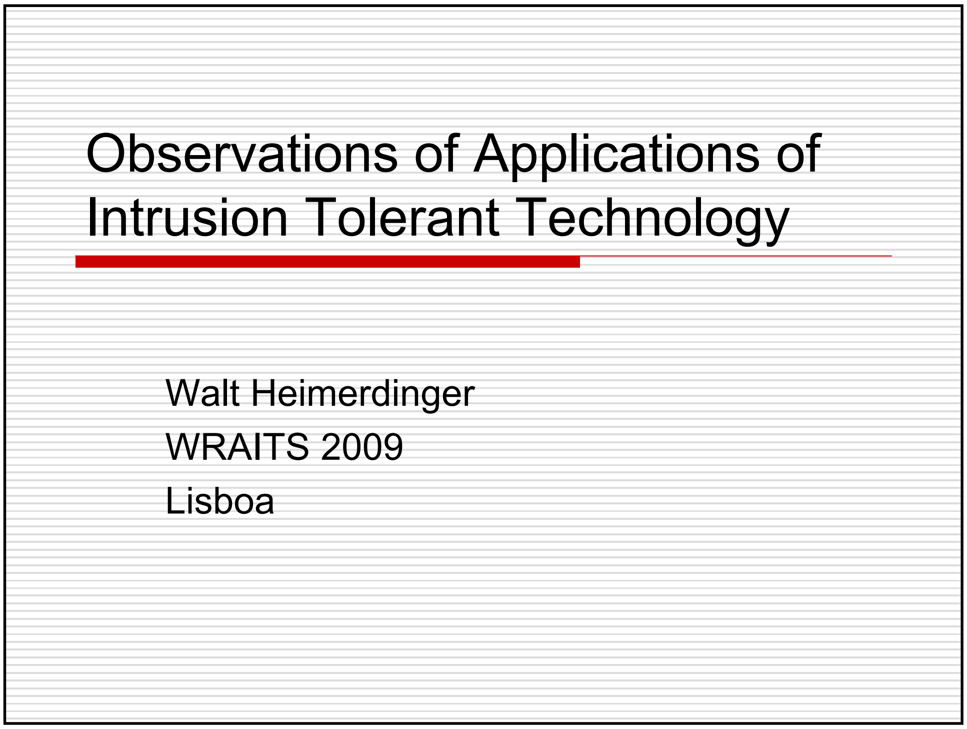
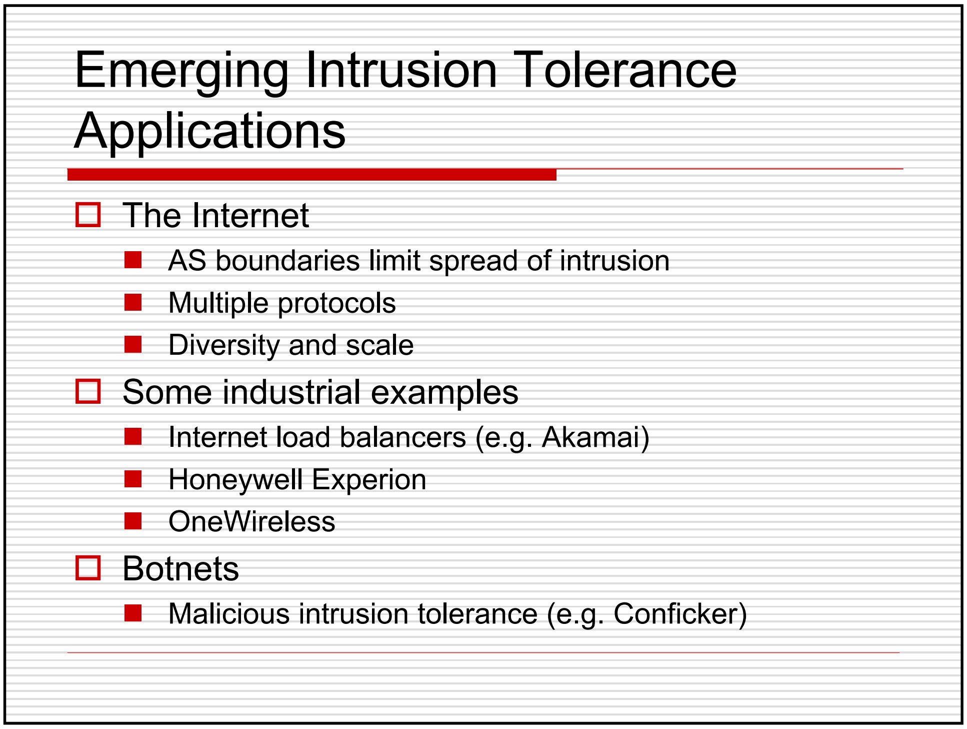


Observations of Applications of Intrusion Tolerant Technology

Walt Heimerdinger
WRAITS 2009
Lisboa



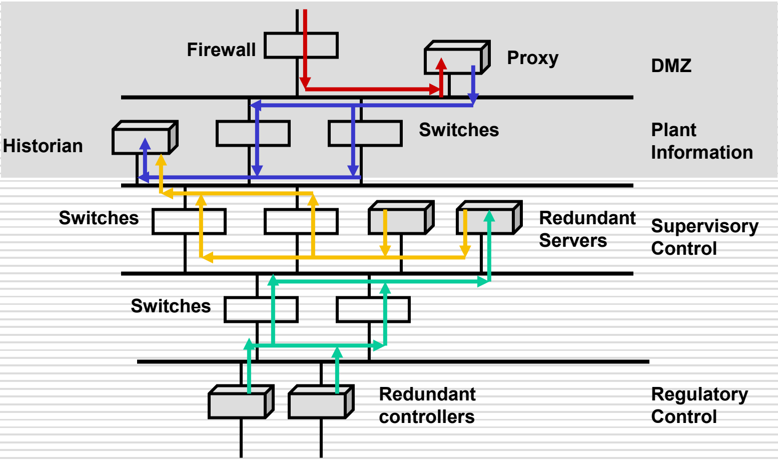
Emerging Intrusion Tolerance Applications

- The Internet
  - AS boundaries limit spread of intrusion
  - Multiple protocols
  - Diversity and scale
- Some industrial examples
  - Internet load balancers (e.g. Akamai)
  - Honeywell Experion
  - OneWireless
- Botnets
  - Malicious intrusion tolerance (e.g. Conficker)

Experion Intrusion Tolerance

- Extends existing fault tolerance architecture
  - Redundant computers and communications
  - Fail silent fault detection
- Adds multiple layers of defense
  - Proxies provide outside access to internal process data
  - VLANs insure that intrusions at outer layers have no path to applications at inner layers

Experion Defense in Depth



Firewall

Proxy

DMZ

Historian

Switches

Plant Information

Switches

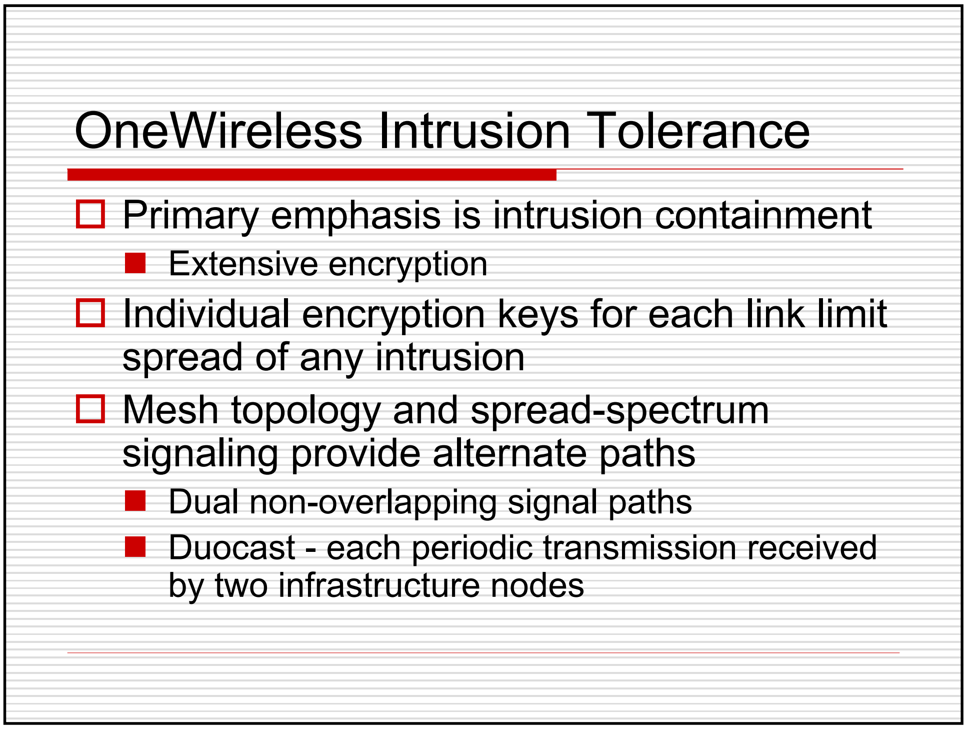
Redundant Servers

Supervisory Control

Switches

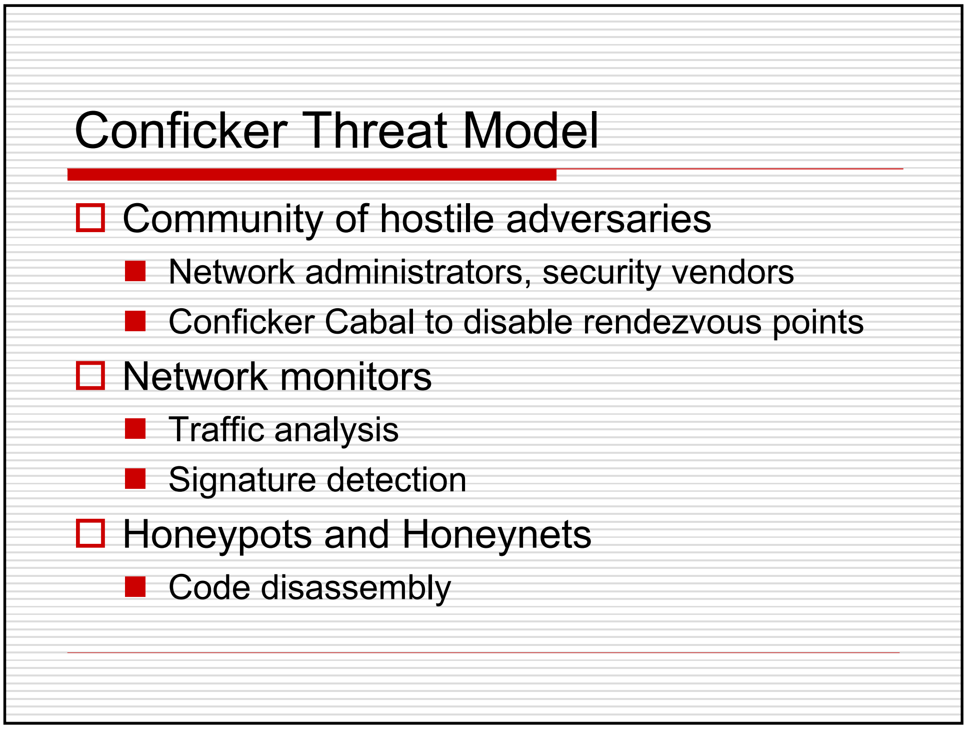
Redundant controllers

Regulatory Control



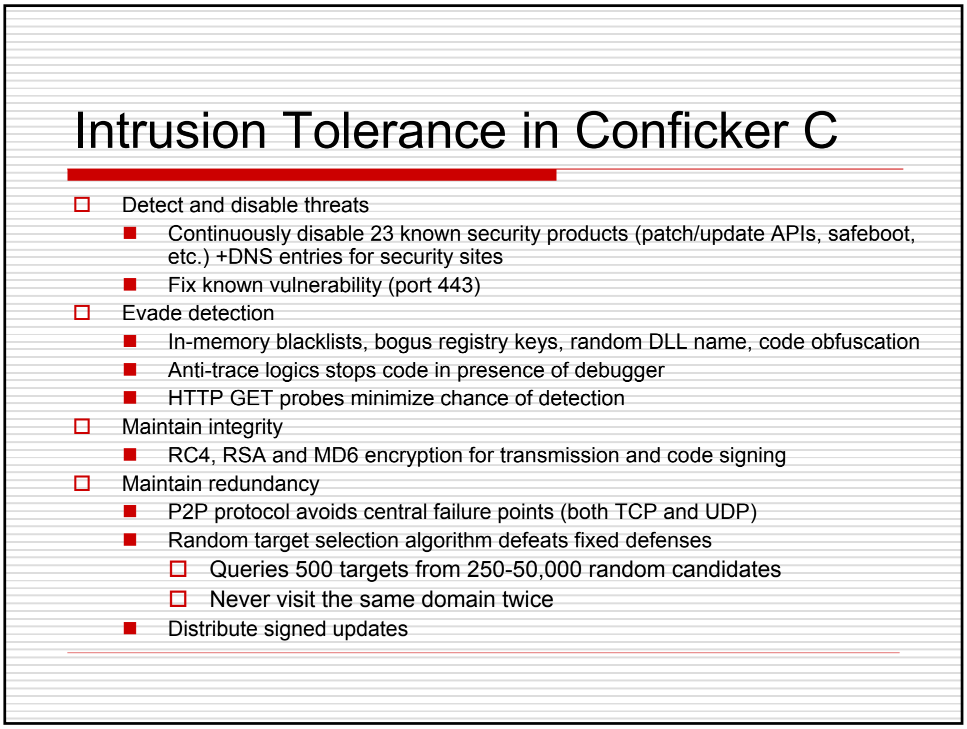
OneWireless Intrusion Tolerance

- □ Primary emphasis is intrusion containment
  - Extensive encryption
- □ Individual encryption keys for each link limit spread of any intrusion
- □ Mesh topology and spread-spectrum signaling provide alternate paths
  - Dual non-overlapping signal paths
  - Duocast - each periodic transmission received by two infrastructure nodes



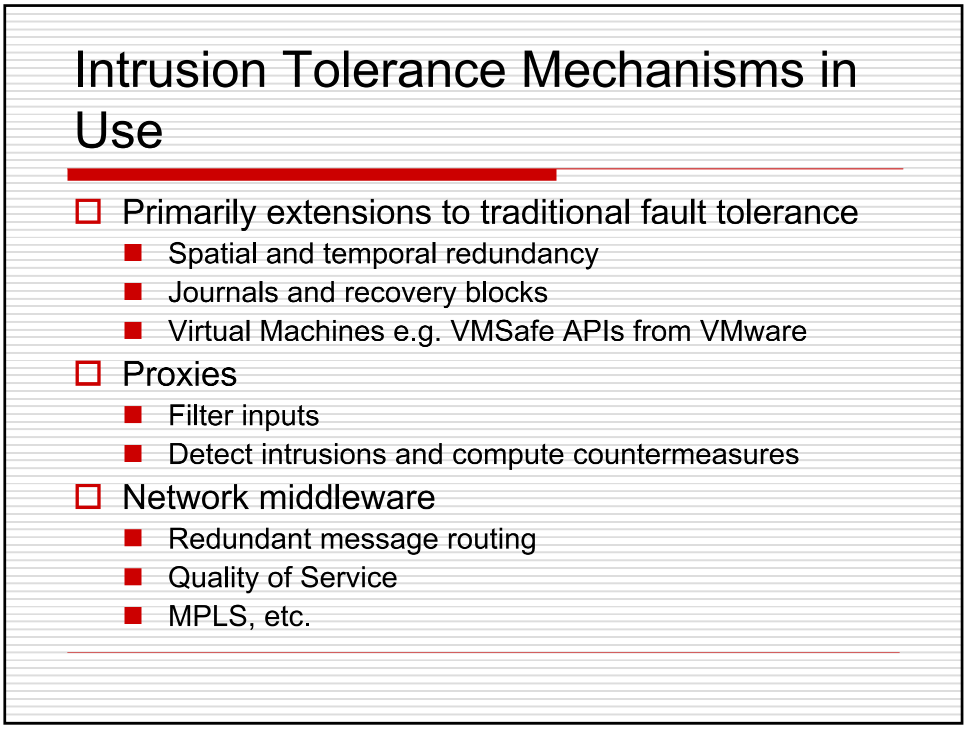
Conficker Threat Model

- □ Community of hostile adversaries
  - Network administrators, security vendors
  - Conficker Cabal to disable rendezvous points
- □ Network monitors
  - Traffic analysis
  - Signature detection
- □ Honeypots and Honeynets
  - Code disassembly



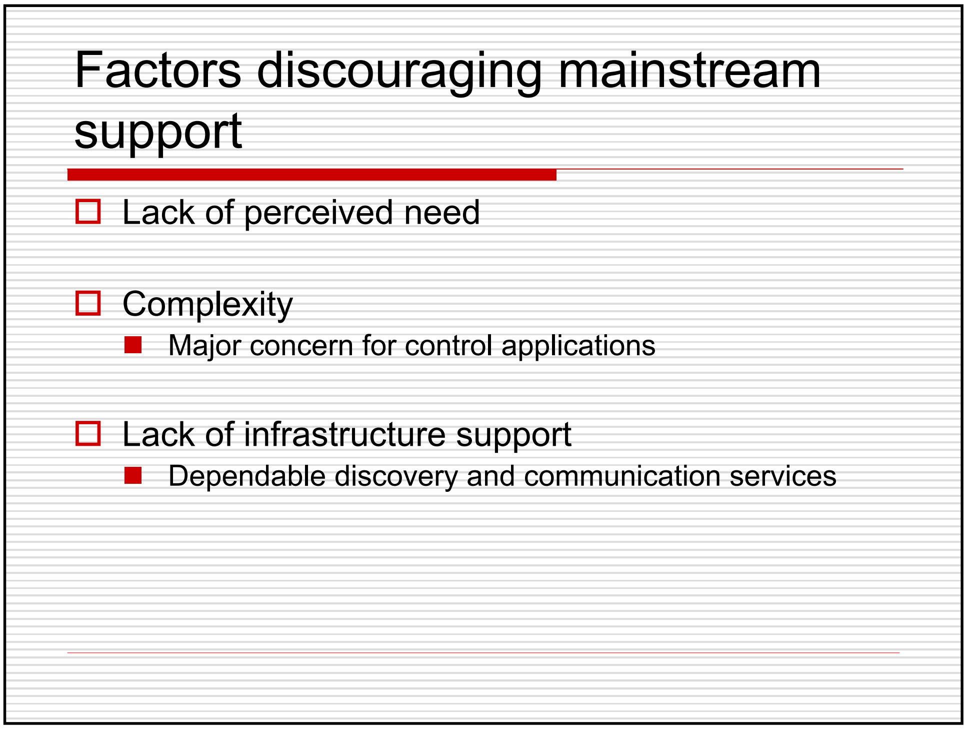
Intrusion Tolerance in Conficker C

- Detect and disable threats
  - Continuously disable 23 known security products (patch/update APIs, safeboot, etc.) +DNS entries for security sites
  - Fix known vulnerability (port 443)
- Evade detection
  - In-memory blacklists, bogus registry keys, random DLL name, code obfuscation
  - Anti-trace logics stops code in presence of debugger
  - HTTP GET probes minimize chance of detection
- Maintain integrity
  - RC4, RSA and MD6 encryption for transmission and code signing
- Maintain redundancy
  - P2P protocol avoids central failure points (both TCP and UDP)
  - Random target selection algorithm defeats fixed defenses
    - Queries 500 targets from 250-50,000 random candidates
    - Never visit the same domain twice
  - Distribute signed updates



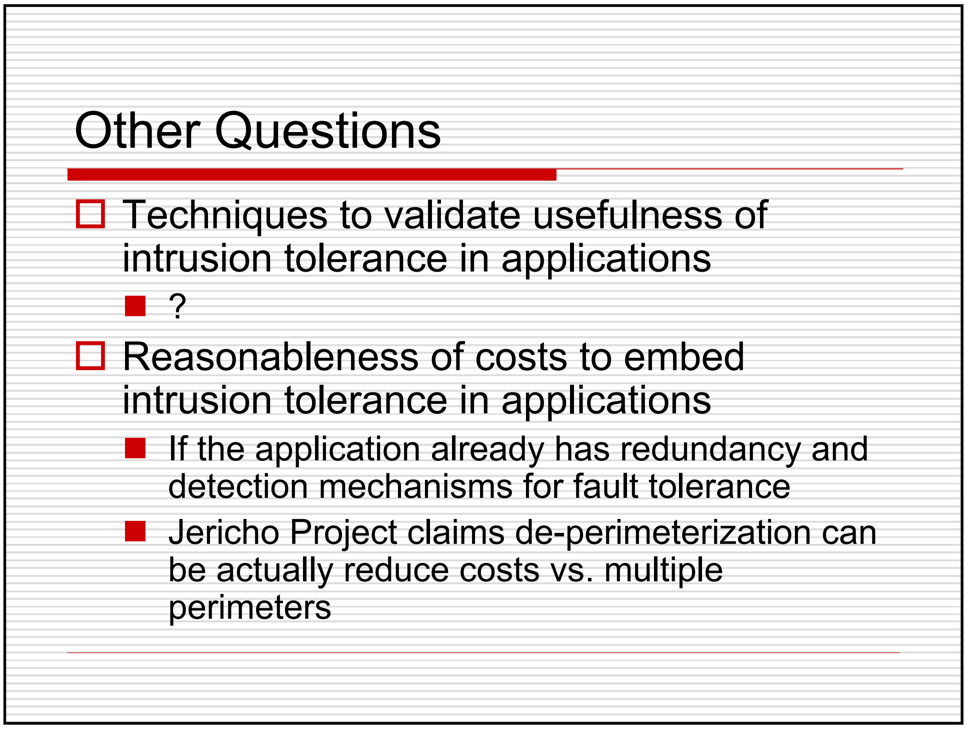
Intrusion Tolerance Mechanisms in Use

- Primarily extensions to traditional fault tolerance
  - Spatial and temporal redundancy
  - Journals and recovery blocks
  - Virtual Machines e.g. VMSafe APIs from VMware
- Proxies
  - Filter inputs
  - Detect intrusions and compute countermeasures
- Network middleware
  - Redundant message routing
  - Quality of Service
  - MPLS, etc.



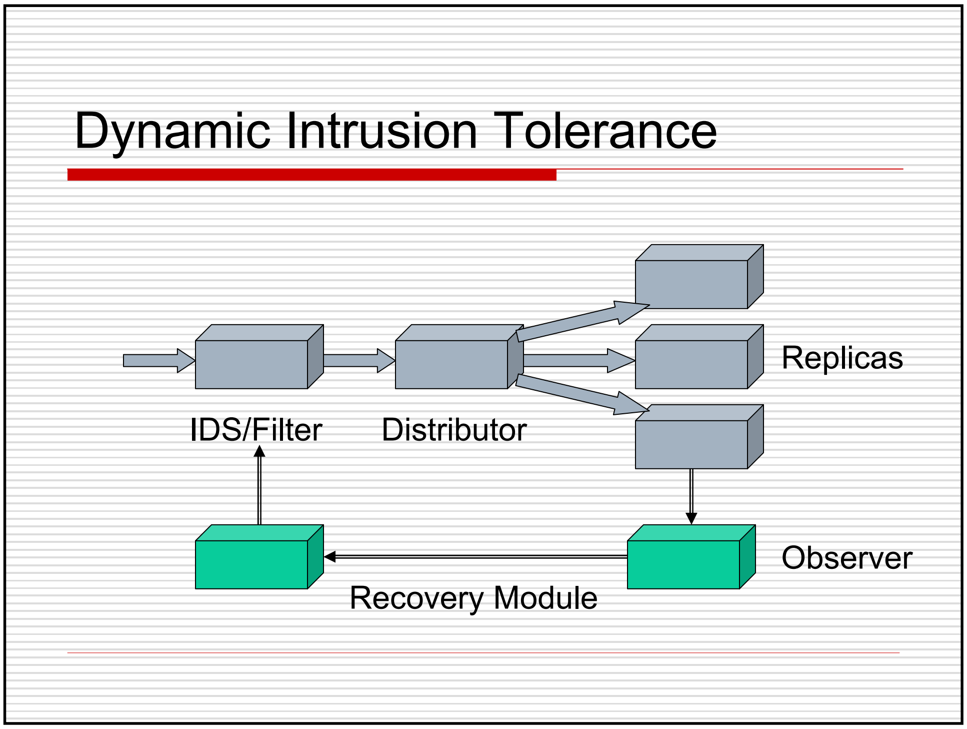
Factors discouraging mainstream support

- Lack of perceived need
- Complexity
  - Major concern for control applications
- Lack of infrastructure support
  - Dependable discovery and communication services



Other Questions

- Techniques to validate usefulness of intrusion tolerance in applications
  - ?
- Reasonableness of costs to embed intrusion tolerance in applications
  - If the application already has redundancy and detection mechanisms for fault tolerance
  - Jericho Project claims de-perimeterization can be actually reduce costs vs. multiple perimeters



Dynamic Intrusion Tolerance

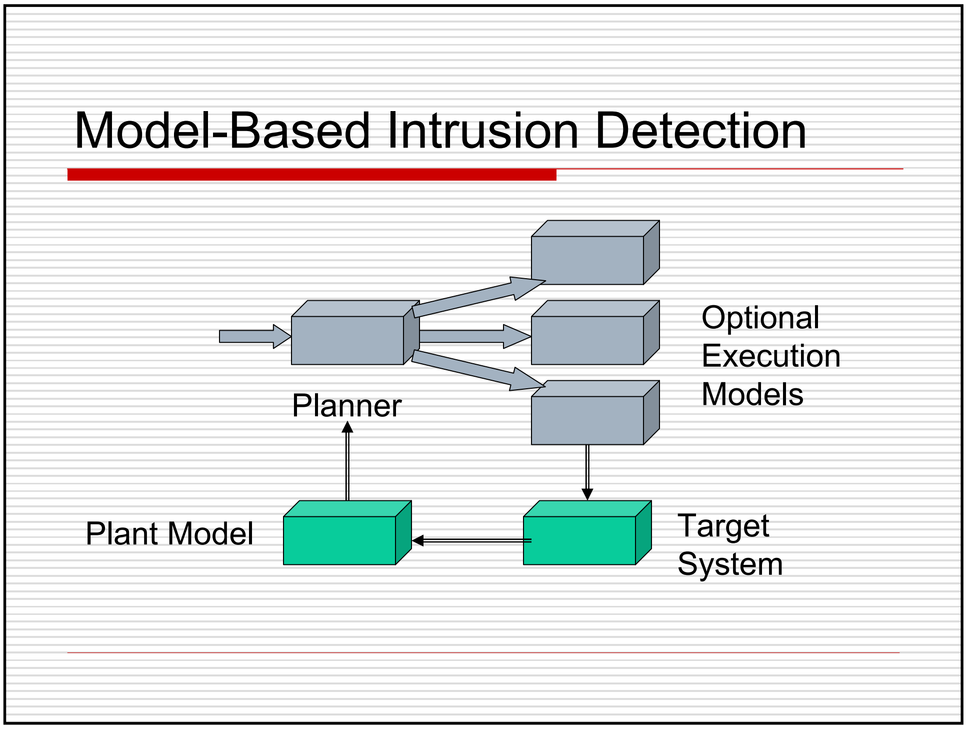
Replicas

IDS/Filter

Distributor

Observer

Recovery Module



Model-Based Intrusion Detection

Optional
Execution
Models

Planner

Plant Model

Target
System