Organically Assured & Survivable Information Systems

2 April 2003

http://www.darpa.mil/ipto/research/oasis/index.html

http://www.tolerantsystems.org

Operate Through Attacks!!

Dr. Jaynarayan Lala

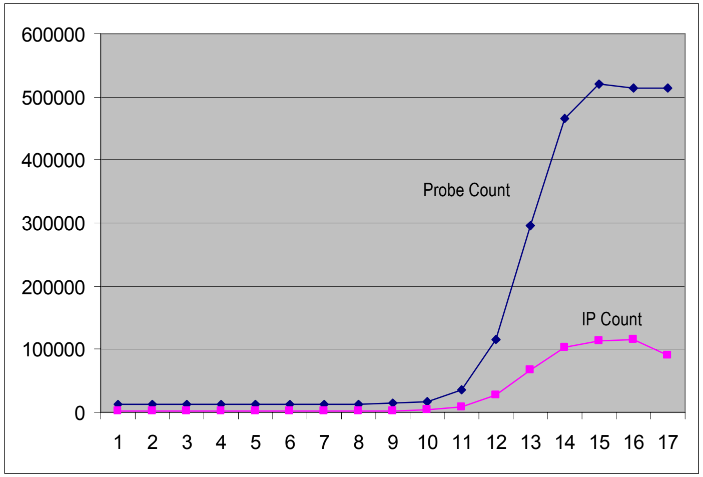
Program Manager

Information Processing Technology Office

Cyber Attacks

Reality

- Code Red Worm*
  - –Code Red I - July 17, 2001; Code Red II - August 4, 2001
  - –Exploits vulnerability in Microsoft's IIS Web Server software
  - –Performed a DOS attack against www.whitehouse.gov.
  - –Relatively benign payload. Defaces web sites.
  - –Infected 250,000 systems in 9 hours; 975,000 total

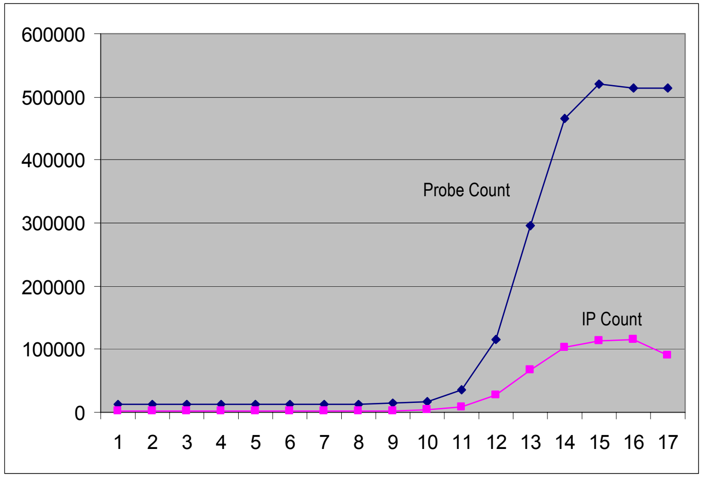


*GAO Report GAO-01-1073T of 29 August 2001

Cyber Attacks

Reality

- Code Red Worm*
  - –Code Red I - July 17, 2001; Code Red II - August 4, 2001
  - –Exploits vulnerability in Microsoft's IIS Web Server software
  - –Performed a DOS attack against www.whitehouse.gov.
  - –Relatively benign payload. Defaces web sites.
  - –Infected 250,000 systems in 9 hours; 975,000 total



*GAO Report GAO-01-1073T of 29 August 2001

Imaginable

- Andy Warhol Worm
  - –Spreads throughout internet in 15 minutes
  - –Malicious payload, such as the Nimda virus
    - –Provides remote attackers "Administrator" privileges and access to entire file system

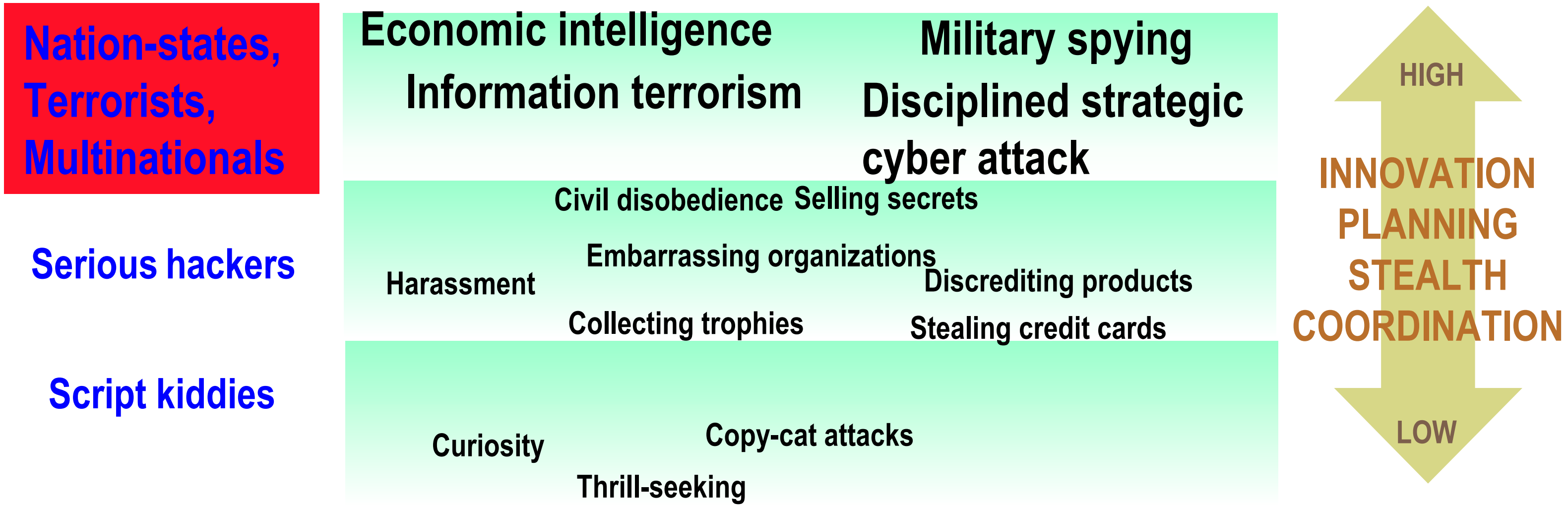
Sapphire/Slammer Worm

- Sapphire/Slammer worm recently affected Microsoft SQL servers.
- Required roughly 10 minutes to spread worldwide
- At its peak, Sapphire scanned the Internet at over 55 million IP addresses/second, causing major disruptions on the net*

* http://www.silicondefense.com/sapphire/

What was only imaginable a year ago, is now a reality!

Defending Against the Most Serious Attacks



Nation-states, Terrorists, Multinationals

Economic intelligence

Information terrorism

Military spying

Disciplined strategic cyber attack

Serious hackers

Civil disobedience

Selling secrets

Embarrassing organizations

Harassment

Discrediting products

Collecting trophies

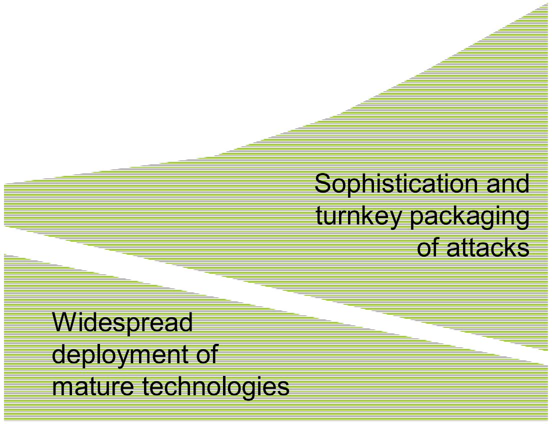
Stealing credit cards

Script kiddies

Curiosity

Copy-cat attacks

Thrill-seeking



Sophistication and turnkey packaging of attacks

Widespread deployment of mature technologies

Increased population of attackers and access to damaging attacks

Reduced opportunities to attack DOD systems

The Critical IW Attack Problem

- Still face high volume of harassment attacks
- Nation-state-level threats may use harassment attacks as cover, diversion, or disguise
- Determination and attribution of IW attacks is critical

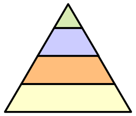
Intrusion Tolerance: A New Paradigm for Security

Prevent Intrusions
(Access Controls, Cryptography, Trusted Computing Base)

Trusted Computing Base

Access Control & Physical Security

Cryptography



Multiple Security Levels

1st Generation: Protection

Intrusion Tolerance: A New Paradigm for Security

Prevent Intrusions
(Access Controls, Cryptography, Trusted Computing Base)

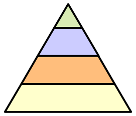
But intrusions will occur

Detect Intrusions, Limit Damage
(Firewalls, Intrusion Detection Systems, Virtual Private Networks, PKI)

Trusted Computing Base

Access Control & Physical Security

Cryptography

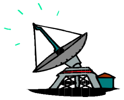


Multiple Security Levels

1st Generation: Protection

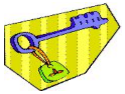
Firewalls

Boundary Controllers



Intrusion Detection Systems

VPNs



PKI

2nd Generation: Detection

Intrusion Tolerance: A New Paradigm for Security

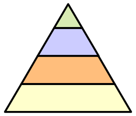
Prevent Intrusions
(Access Controls, Cryptography, Trusted Computing Base)

But intrusions will occur

Trusted Computing Base

Access Control & Physical Security

Cryptography



Multiple Security Levels

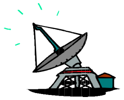
1st Generation: Protection

Detect Intrusions, Limit Damage
(Firewalls, Intrusion Detection Systems, Virtual Private Networks, PKI)

But some attacks will succeed

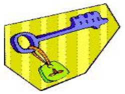
Firewalls

Boundary Controllers



Intrusion Detection Systems

VPNs



PKI

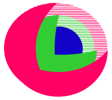
2nd Generation: Detection

Tolerate Attacks
(Redundancy, Diversity, Deception, Wrappers, Proof-Carrying Code, Proactive Secret Sharing)

Intrusion Tolerance

Big Board View of Attacks
Real-Time Situation Awareness
& Response

Graceful Degradation



Hardened Operating System

3rd Generation: Tolerance

Information Assurance Attributes*

- **Integrity**
  - Maintain data and program integrity in the face of intrusions and malicious faults.
- **Availability**
  - Counter Denial-of-Service attacks and maintain high system availability.
- **Confidentiality**
  - Prevent unauthorized disclosure of information.
- **Authentication**
  - Prevent unauthorized access.
- **Non-repudiation**
  - Method by which the sender of data is provided with proof of delivery and the recipient is assured of the sender's identity, so that neither can later deny having processed the data.

* Joint Pub 3-13 "Joint Doctrine for Information Operations"

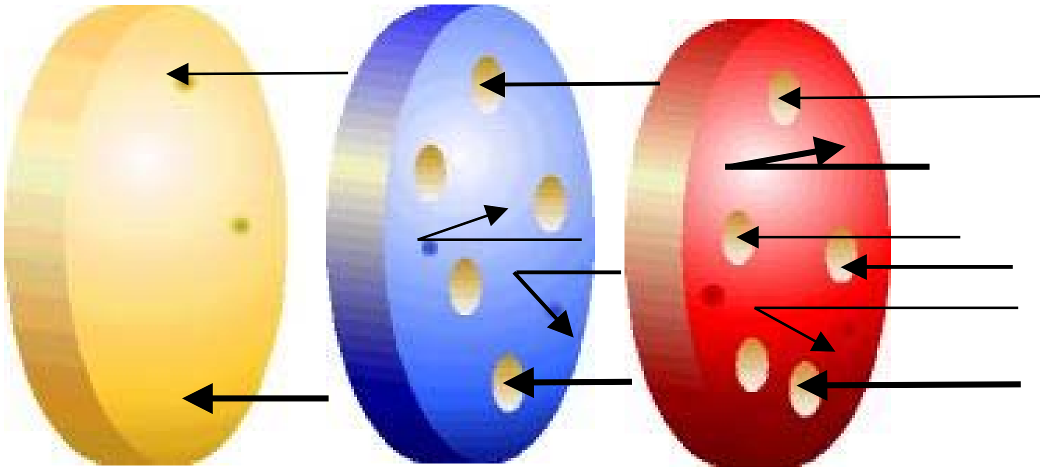
OASIS Approach & Challenges

TECHNICAL APPROACH

Confine malicious code-- compare actual behavior with predicted

Detect errors: watermark, time/value domain anomalies, rear guards

Error compensation and recovery: distributed computation, design diversity & deception



ERROR DETECTION / TOLERANCE TRIGGERS

CYBER ATTACKS

ERROR COMPENSATION / RESPONSE / RECOVERY

EXECUTION MONITORS

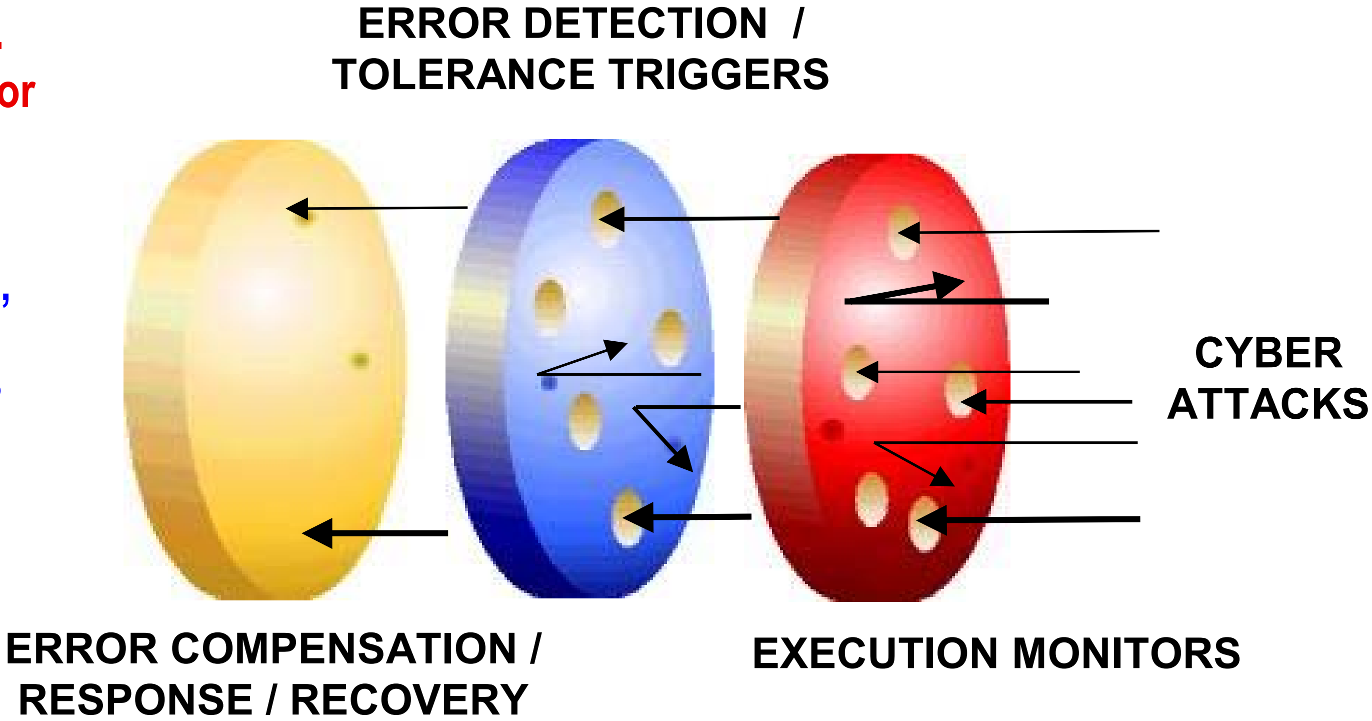
OASIS Approach & Challenges

TECHNICAL APPROACH

Confine malicious code--compare actual behavior with predicted

Detect errors: watermark, time/value domain anomalies, rear guards

Error compensation and recovery: distributed computation, design diversity & deception


ERROR DETECTION / TOLERANCE TRIGGERS
CYBER ATTACKS
ERROR COMPENSATION / RESPONSE / RECOVERY
EXECUTION MONITORS

TOP TECHNICAL CHALLENGES

Real-time trade of security, performance & functionality

Cost-effective solutions

Validation and verification

OASIS Approach & Challenges

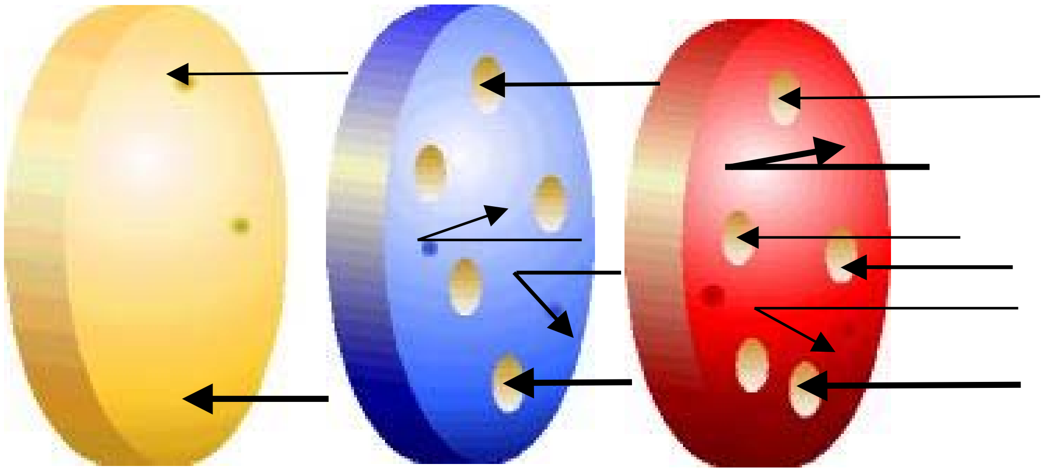
TECHNICAL APPROACH

Confine malicious code-- compare actual behavior with predicted

Detect errors: watermark, time/value domain anomalies, rear guards

Error compensation and recovery: distributed computation, design diversity & deception

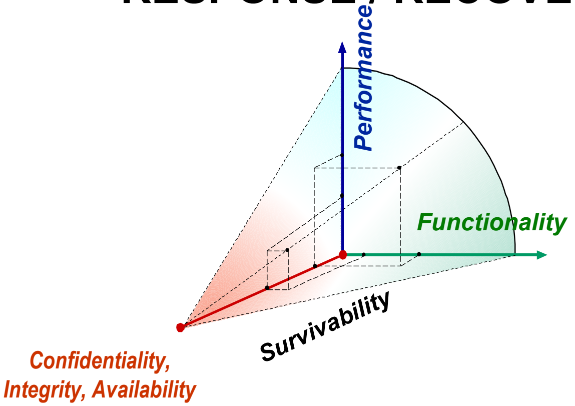
ERROR DETECTION / TOLERANCE TRIGGERS



CYBER ATTACKS

ERROR COMPENSATION / RESPONSE / RECOVERY

EXECUTION MONITORS

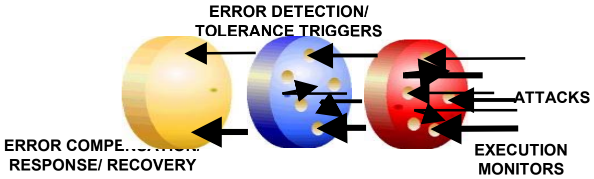


TOP TECHNICAL CHALLENGES

Real-time trade of security, performance & functionality

Cost-effective solutions

Validation and verification



OASIS Technologies

ERROR COMPENSATION/ RESPONSE/ RECOVERY

Spatial, Temporal, Design, and Analytical Redundancies, Dynamic Reconfiguration, Quality of Service Trade-Offs, Fragmentation & Dispersal, Deception (Randomness, Uncertainty, Agility, Stealth), Graceful Degradation, Intrusion Tolerant Architectures

ERROR DETECTION/ TOLERANCE TRIGGERS

Watermarks, Mediated Interfaces, Rear Guard, Value & Time Domain Error Detectors, Comparison & Voting, Acceptance Checks, Redundancy-Based Cyber Attack Detection

EXECUTION MONITORS

In-Line Reference Monitors, Sandbox Active Scripts, Code Interposition, Wrappers, Proof Carrying Code, Graph Based Program Encoding, Monitor COTS Binaries, Secure Mobile Code Format, Operate through Mobile/ Malicious Code Attack

FAULT AVOIDANCE

Provably Correct Protocols, Secure-design Principles, Software Vulnerability Detection, Design Assessment and Validation

OASIS Projects

DARPA

		Performer	Organization	Project
Error Detection/Tolerance Triggers	Error Compensation/Response/Recovery	Prof. Andrew Chien	UCSD	Agile Objects: Component-based Inherent Survivability
		Prof. Pradeep Khosla	CMU	Perpetually Available and Secure Information Systems
		Dr. Jim Just	Teknowledge	Hierarchical Adaptive Control for QoS Intrusion Tolerance (HACQIT)
		Dr. Peng Liu	UCMBC	Engineering a Distributed Intrusion Tolerant Database System Using COTS Components
		Dr. Alexander Wolf	Univ. of Colorado	Tolerating Intrusions Through Secure System Reconfiguration
		Dr. Feiyi Wang	MCNC	Scalable Intrusion Tolerant Architecute (SITAR)
		Mr. Alfonso Valdes	SRI, International	Dependable Intrusion Tolerance
		Dr. Dick O'Brien	SCC	Intrusion Tolerant Server Infrastructure
		Dr. Partha Pal	BBN	Intrusion Tolerance by Unpredictable Adaptation
		Ms. Janet Lepanto	Draper	Intrusion Tolerance Using Masking, Redundancy and Dispersion
		Mr. Lee Badger	NAI Lab	Self-Protecting Mobile Agents
		Mr. Gregg Tally	NAI Lab	Intrusion Tolerant Distributed Object Systems
	Execution Monitors	Dr. Gary McGraw	Cigital	An Investigation of Extensible System Security for Highly Resource-Constrained Wireless Devices
		Dr. Robert Balzer	Teknowledge	Integrity Through Mediated Interfaces
		Prof. Anant Agarwal	InCert	A Binary Agent Technology for COTS Software Integrity
		Dr. Robert Balzer	Teknowledge	Enterprise Wrappers for Information Assurance(NT)
		Mr. Mark Feldman	NAI Lab	Enterprise Wrappers for Information Assurance (Unix)
		Prof. Andrew Appel	Princeton	Scaling Proof-Carrying Code to Production Compilers and Security Policies
		Prof. Fred Schneider	Cornell	Containment and Integrity for Mobile Code
	Fault Avoidance	Dr. Gary McGraw	Cigital	An Aspect Oriented Security Assurance Solution
		Prof. Crispin Cowan	WireX	Autonomix: Component, System and Network Autonomy
		Dr. Victoria Stavridou	SRI, International	Intrusion Tolerant Software Architecture
		Prof. Michael Franz	UC, Irvine	Reconciling Execution Efficiency With Provable Security
		Dr. Howard Shrobe	MIT	Active Trust Management for Autonomous Adaptive Survivable Systems
		Dr. Ranga Ramanujan	ATC	Randomized Failover Intrusion Tolerant Systems (RFITS)

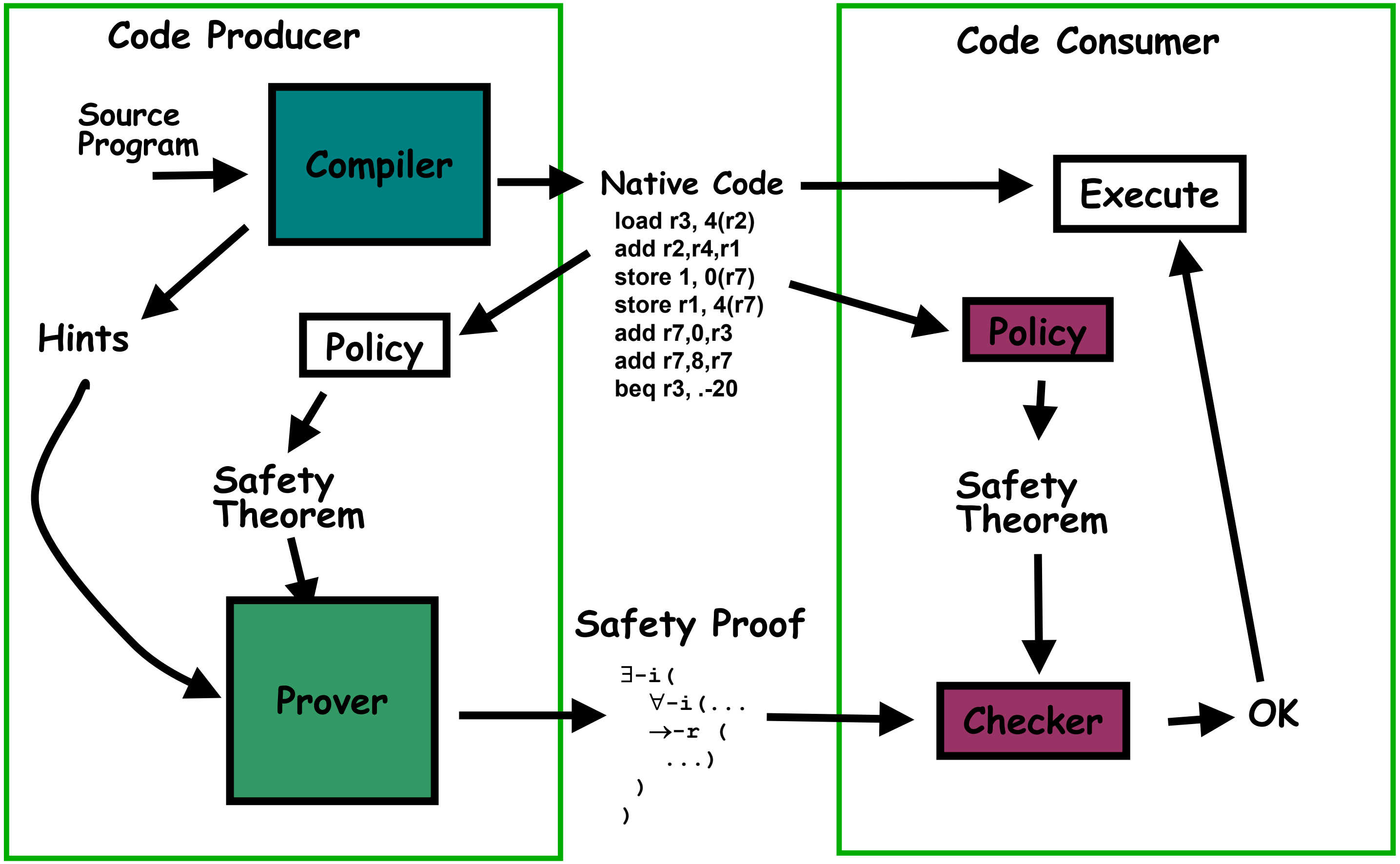
Number of Projects Started Under OASIS: 39

Number of OASIS Projects Active Today: 25

INFORMATION PROCESSING TECHNOLOGY OFFICE

Proof-carrying Code

- **Princeton/Intel collaboration**
  - PCC Technology being applied to Intel's "Just in Time" compiler for Microsoft's Common Language Runtime (CLR).
  - Demonstrated scalable certifying compiler that produces proof of program behavior along with the code.
- **Princeton University (Prof. Andrew Appel)**
- **Yale University (Prof. Zhong Shao)**



Code Producer

Source Program

Compiler

Hints

Policy

Safety Theorem

Prover

Native Code

```
load r3, 4(r2)
add r2,r4,r1
store 1, 0(r7)
store r1, 4(r7)
add r7,0,r3
add r7,8,r7
beq r3, .-20
```

Safety Proof

```
∃-i(
   ∀-i(...
   →-r (
     ...)
  )
)
```

Code Consumer

Execute

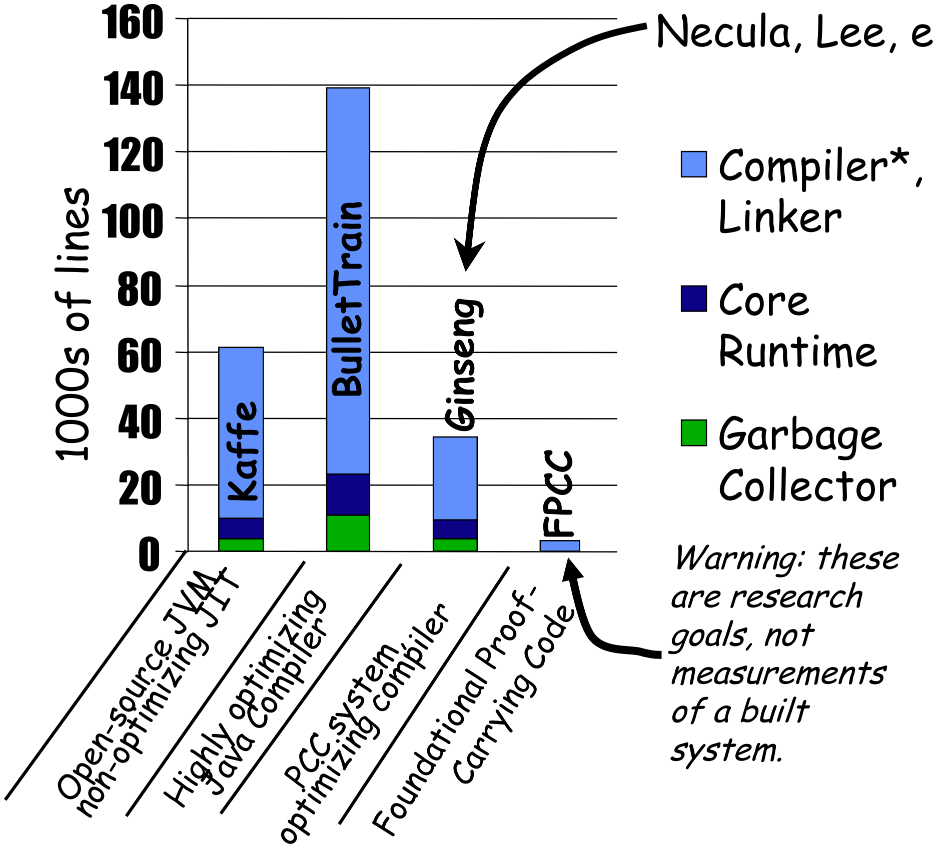
Policy

Safety Theorem

Checker

OK

Proof-carrying Code



Necula, Lee, et al.

Warning: these are research goals, not measurements of a built system.

Size of Trusted Computing Base

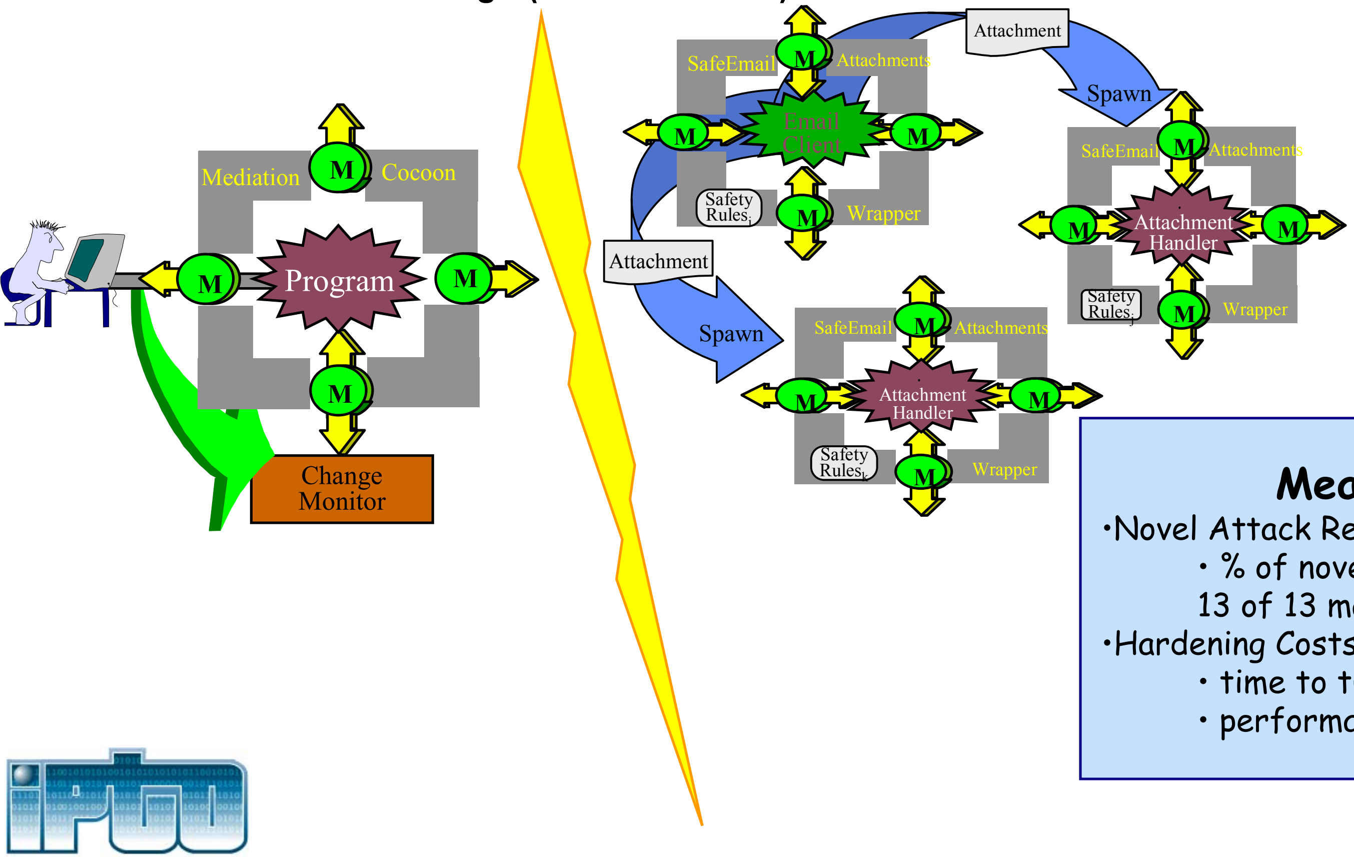
Measures of Merit

Goal:

- Reduce size of Trusted Computing Base to 4K Source Lines of Code
  - Approximately 10% of comparable functionality PCC compiler
- Actual TCB size achieved
  - 3K SLOC
  - 25% better than a very aggressive goal

Safe E-mail Wrappers

- **Transitioning to PACOM for scalability tests and experience in military operational environment**
  - Demonstrated protection against mobile malicious code (malicious email attachments, scripts in email bodies, web applets, active-x controls, downloaded programs), corrupted executables and documents, and latent flaws in applications by several different techniques
  - Not signature based; techniques work on novel viruses without any customization
- **Teknowledge (Dr. Bob Balzer)**



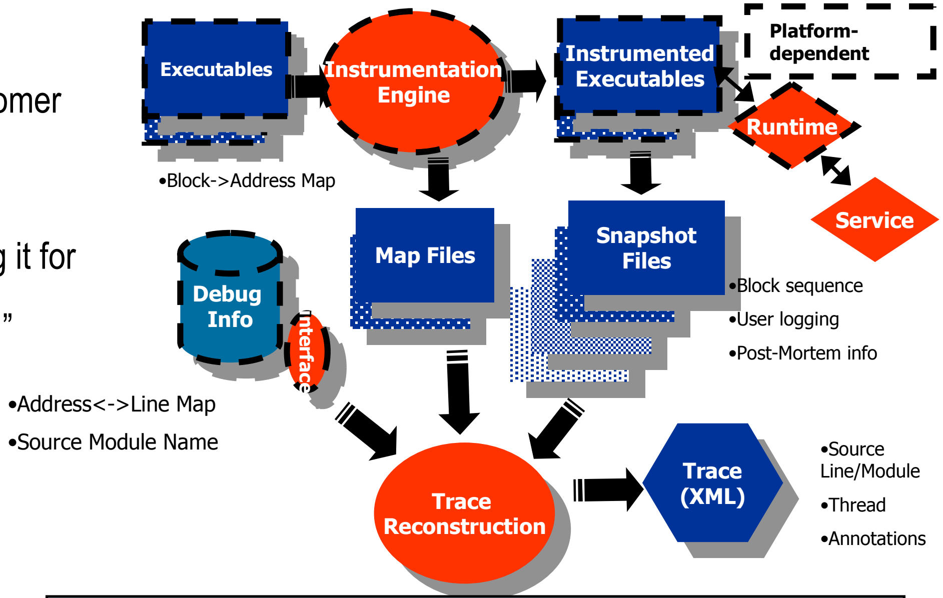
Measures of Merit

- Novel Attack Resistance:
  - % of novel attacks prevented (detected 13 of 13 malicious attacks)
- Hardening Costs:
  - time to tune security policies (3 -5 days)
  - performance degradation (7% overhead)

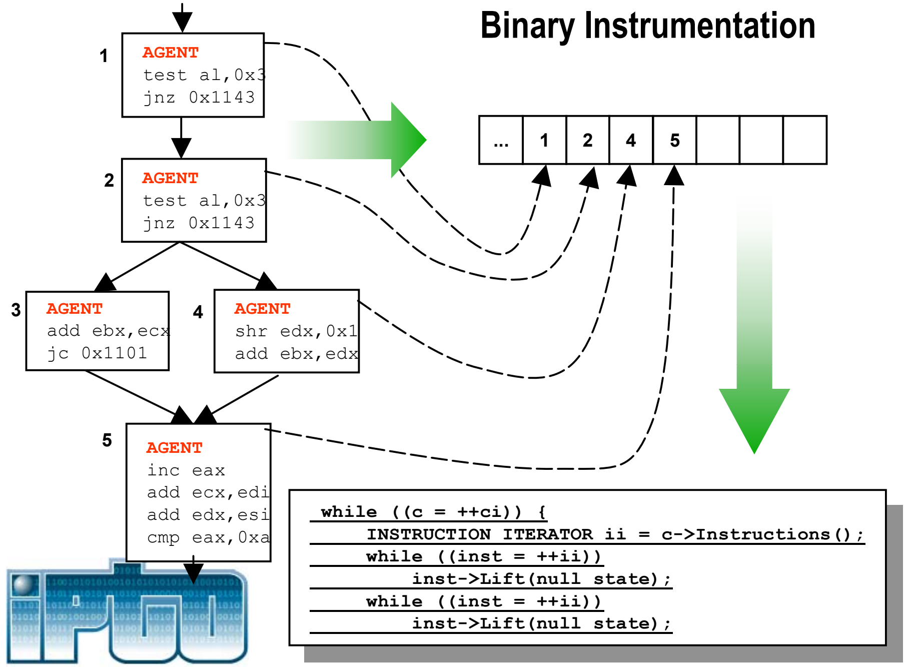
Binary Agents
(InCert Technologies, Dr. Anant Agarwal)

- Halo to a Major Industry Power Systems Control Software
  - Halo monitors, pinpoints, reports on and provides a root cause diagnosis of software faults.
  - Halo is unique in its “always on” capabilities.
  - Monitors applications deployed into production or out to customer sites.
  - Company experienced:
    - Testing cycle was cut in half,
    - In one month went from instrumenting PMCS to preparing it for full production deployment.
    - “It helps us have the most reliable software in our market.”

Major Components



Binary Instrumentation

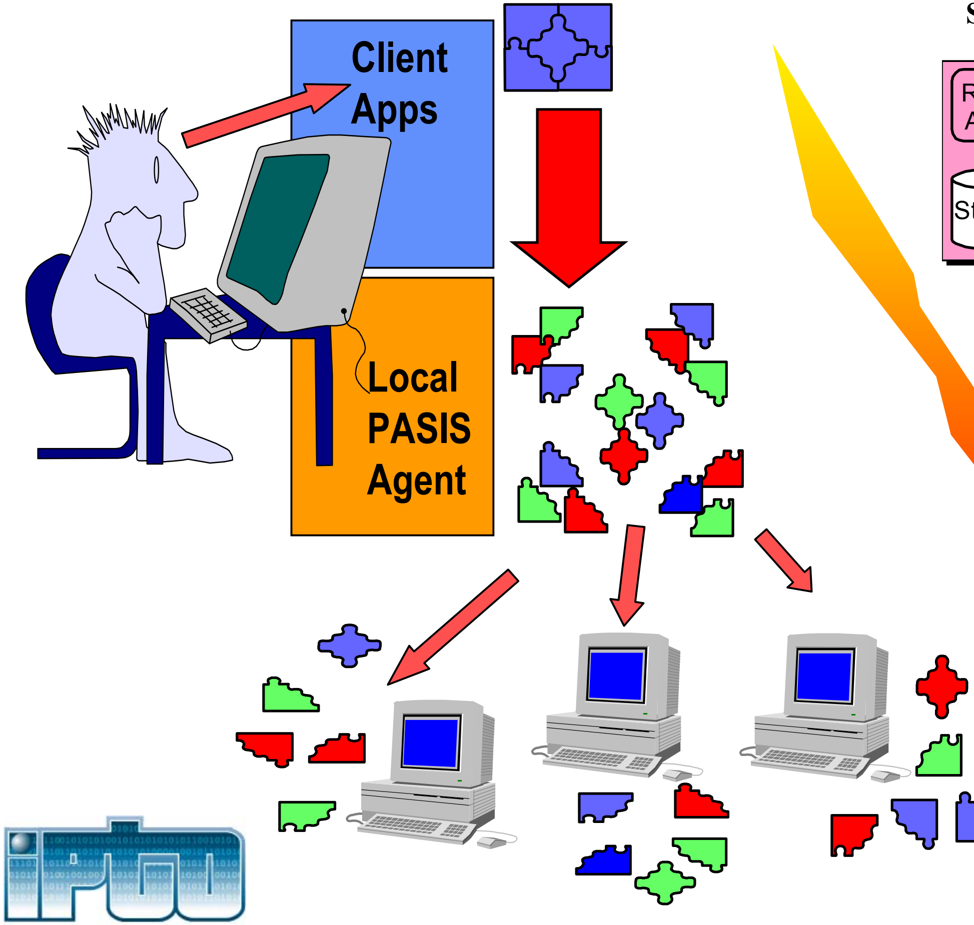


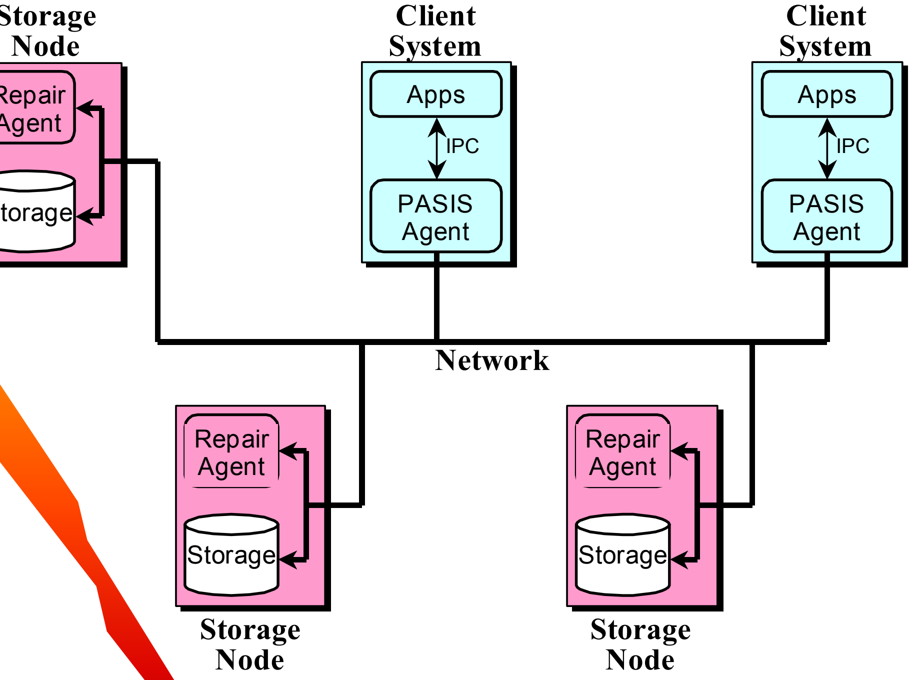
```
while ((c = ++ci)) {
    INSTRUCTION_ITERATOR ii = c->Instructions();
    while ((inst = ++ii))
        inst->Lift(null_state);
    while ((inst = ++ii))
        inst->Lift(null_state);
```

- Percentage of executables successfully instrumented
  - Goal: 100%
  - Accomplished to date: Virtually 100% (approx. 50 real world executables instrumented)
- Performance degradation
  - Goal: less than 5% overhead
  - Accomplished to date: 5-10% overhead when measured in real world scenarios.
- Anomaly detection
  - Goal: 100%
  - Accomplished: Detected 12 of 16 (75%) known problems in field tests.

Intrusion Tolerant Data Storage

- Perpetually Available and Secure Information Systems (PASIS)
- Transitioning to USAF Joint Battlespace Infosphere (JBI) - ***Funded by AFRL***
  - To assure availability, integrity, and confidentiality of JBI "data repository"
  - Demonstrated intrusion tolerant data storage
- Carnegie Mellon University (Prof. Pradeep Khosla)



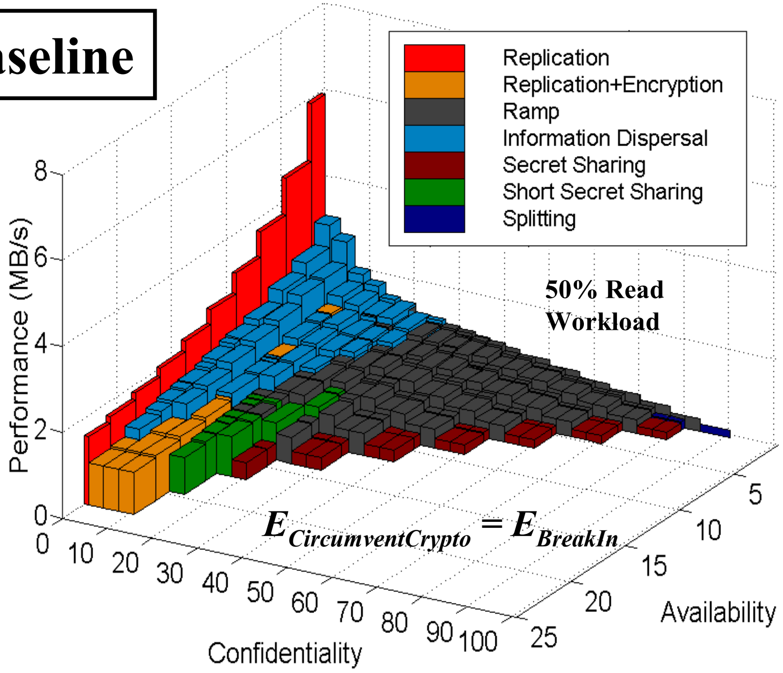


Decentralized Storage Systems

Intrusion Tolerant Data Storage

•PASIS (Performance Trade-offs)

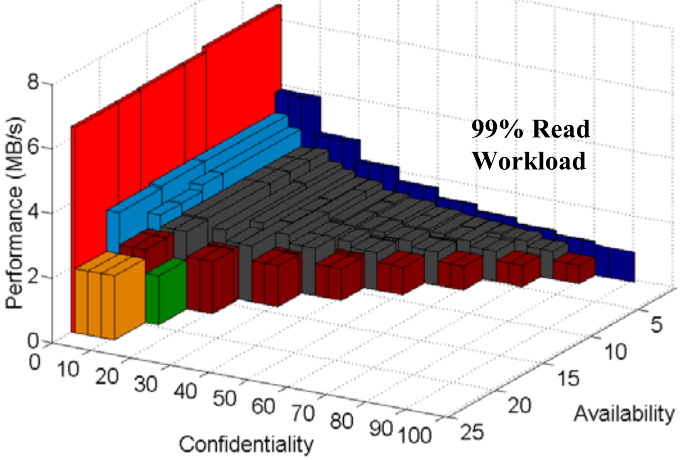
Baseline



50% Read Workload

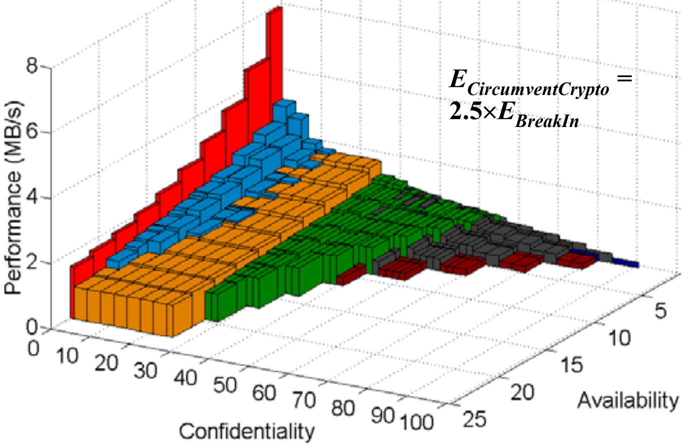
$E_{CircumventCrypto} = E_{BreakIn}$

Extreme Read Workload



99% Read Workload

Security Model Sensitivity



$E_{CircumventCrypto} = 2.5 \times E_{BreakIn}$

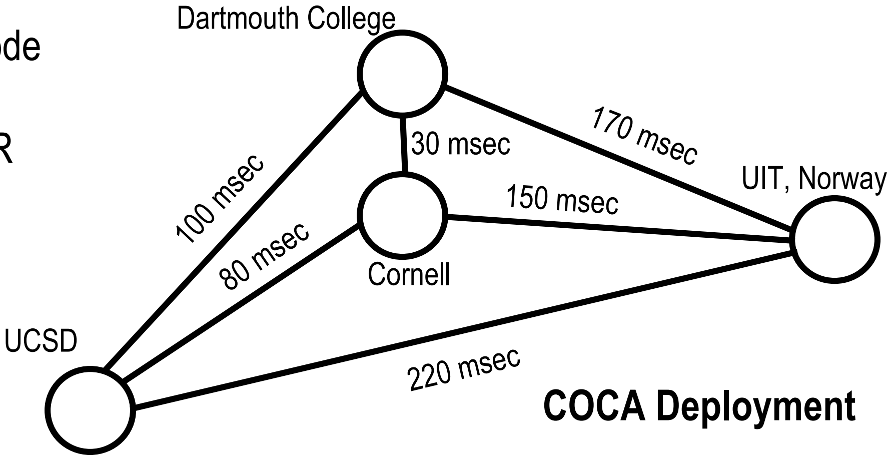
Performance (MB/s)
- •based on simple performance model
- •computed with standard performance eval. techniques

Availability ("nines")
- •standard fault tolerance math with independent failures
- •relative values are useful even if not independent

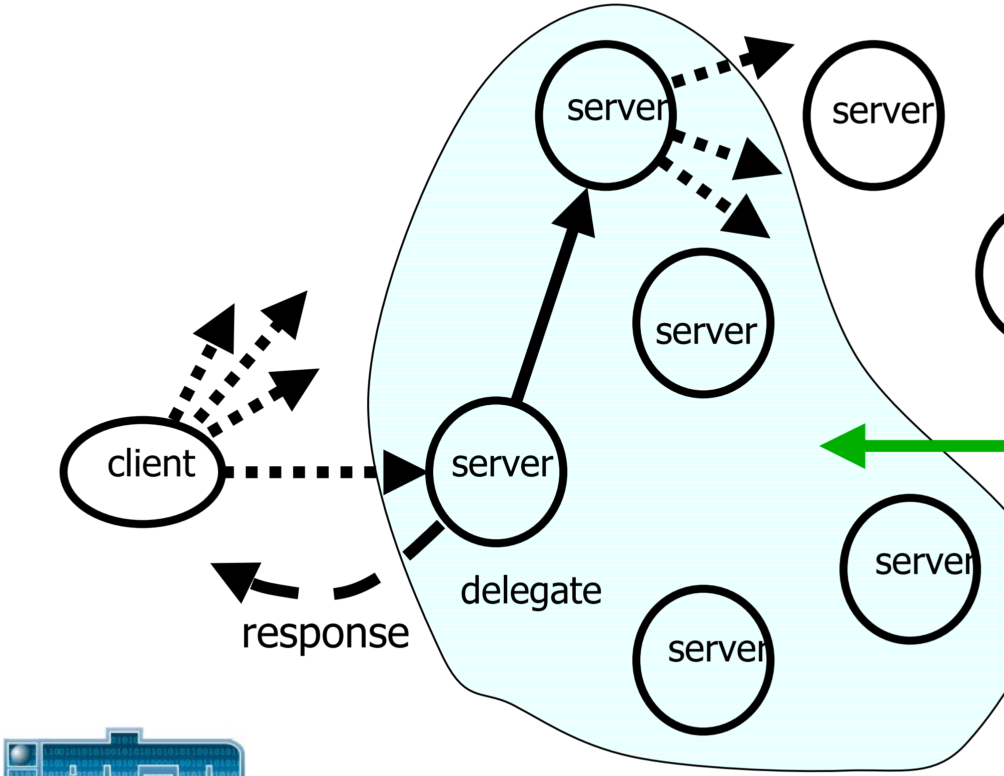
Confidentiality (Effort to compromise)
- •estimate effort involved with possible attack paths
- •overall effort is minimum of possible efforts

Intrusion Tolerant Certificate Authority

- **Prototype implementation:**
  - Approximately 35K lines of new C source code
  - Certificates in accordance with X.509
  - Work Being applied to JBI funded by AFOSR and AFRL
- **Cornell University (Prof. Fred Schneider)**



COCA Deployment



server failure

↓ **disseminated Byzantine quorum**

server compromise

↓ **threshold signature protocol**

mobile attack

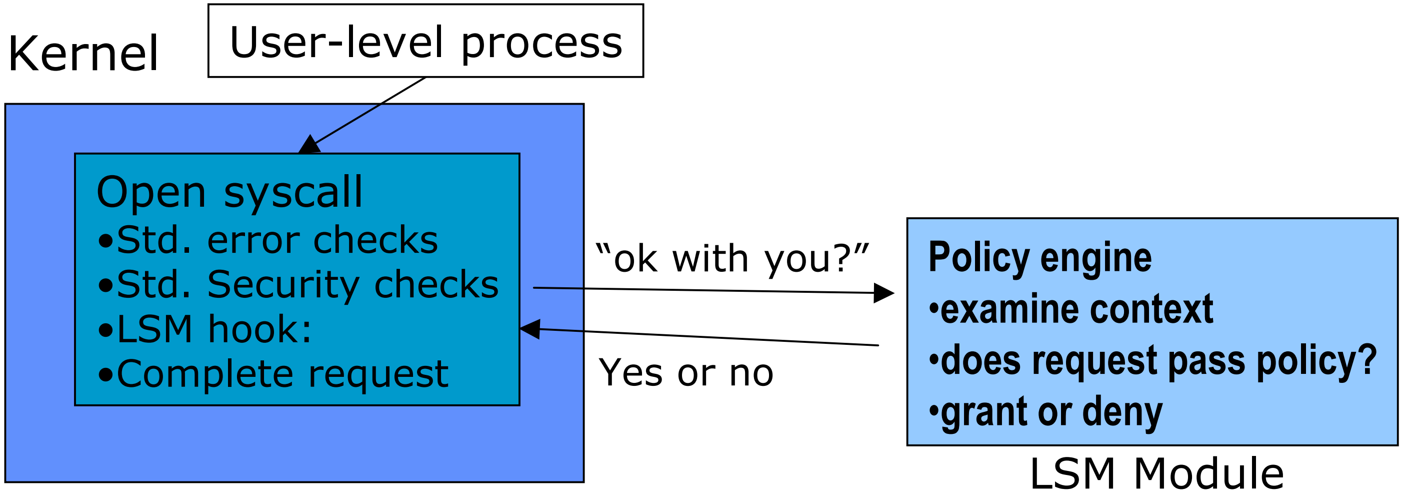
↓ **proactive secret sharing (PSS)**

asynchrony

↓ **asynchronous PSS**

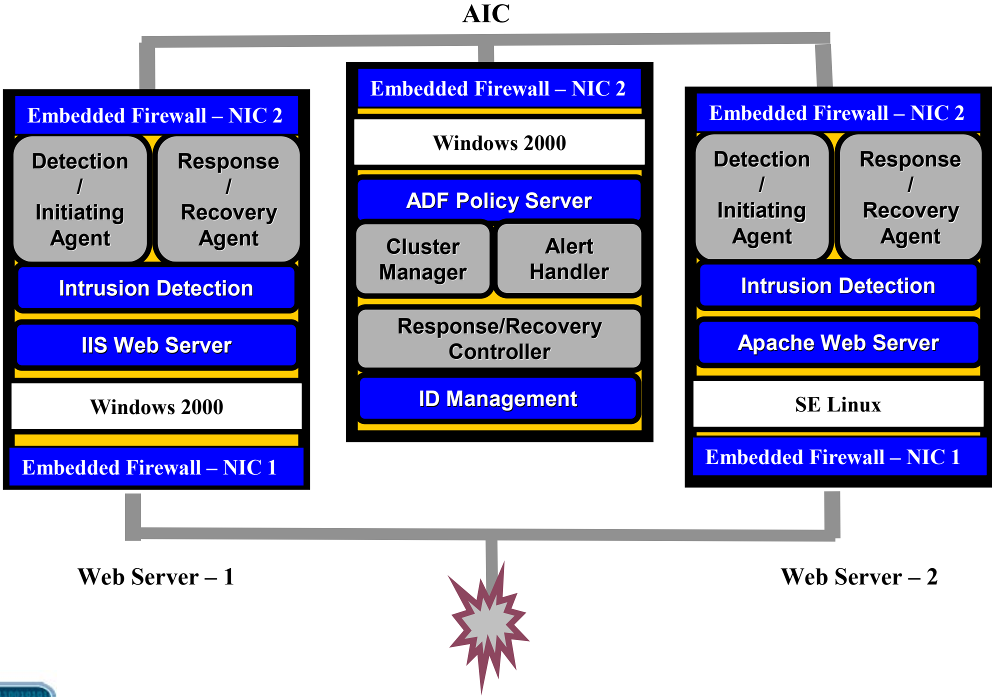
Linux Security Module
(WireX Communications, Dr. Crispin Cowan)

- **LSM design goals are to create a general purpose framework to enable pluggable security modules as an open source security solution for Linux**
  - Allow various security solutions to be employed in the standard Linux kernels.
  - Be general enough to support existing security projects
  - Continue to support root/capabilities, perhaps as a module
- **Linus Torvalds decided to accept LSM into the standard Linux kernel at the June 2002 developer's meeting.**



Intrusion Tolerant Server Architecture

- **Leveraging the commercial success of the Autonomic Distributed Firewall (ADF) to create an intrusion tolerant server architecture**
  - Intrusion tolerant server components: load distribution and network response capability using the ADF Policy Enforcing NICs, server hardening to reduce effectiveness of penetrations, intrusion detection systems that primarily reside on server hosts, an Availability and Integrity Controller (AIC) to manage the system and respond to intrusions reported to it
- **Secure Computing Corporation (Dr. Dick O'Brien)**


AIC
Embedded Firewall – NIC 2
Detection / Initiating Agent
Response / Recovery Agent
Intrusion Detection
IIS Web Server
Windows 2000
Embedded Firewall – NIC 1
Embedded Firewall – NIC 2
Windows 2000
ADF Policy Server
Cluster Manager
Alert Handler
Response/Recovery Controller
ID Management
Embedded Firewall – NIC 2
Detection / Initiating Agent
Response / Recovery Agent
Intrusion Detection
Apache Web Server
SE Linux
Embedded Firewall – NIC 1
Web Server – 1
Web Server – 2

Measures of Merit

- Effectiveness of the approach
  - –Success rate in stopping/recovering from intrusions as measured by red team experiments
  - –Performance overhead as measured by application response time
- Cost/Benefit analysis

Validation Goals

- **In the context of intrusion tolerant technologies, create an underlying scientific foundation that will**
  - measure the effectiveness of novel solutions, and
  - test and evaluate systems in an objective manner.

Validation Challenges

- Unable to specify quantitative assurance requirements.
- Unable to quantitatively state how assured systems and networks are.
- Unable to quantify ability of protective measures to keep out intruders.
- Difficult to characterize capabilities of intrusion detection systems to detect novel attacks.
- Benefits of novel response mechanisms cannot be measured comparatively or absolutely.

An Information Assurance & Survivability Validation Framework

http://www.tolerantsystems.org

Framework Objectives

- Create an information assurance and survivability validation framework that will allow PIs to validate their proposed means for achieving information assurance and survivability
- Continue to organize projects in the OASIS program so that it is possible to
  - Identify to DoD users and DARPA Management where particular technologies and projects can help improve the information assurance and survivability of systems
  - Identify overall coverage of the set of OASIS projects as a whole, so that we can identify vulnerabilities and attacks that are not being addressed
- Use terminology established in the DoD and in the related dependable computing and fault tolerance community (IFIP WG 10.4) for better and wider understanding

Developing a Characterization under the Framework: System/Technology

- **1. A system or more generally a technology has certain functional goals over a domain of application along with certain supporting information assurance and survivability attributes for protection**
  - Examples of functional goals are to provide an application, a database, a mobile code platform, an operating system
  - Domains of application are *where* the technology applies, i.e., to clients, servers, networks, storage, database, middleware, firmware, hardware, etc. and *when* the technology applies, i.e., at design phase, implementation phase, operational phase
  - Information assurance and survivability attributes are standard in the DoD: system availability*, integrity*, confidentiality*, authentication*, and nonrepudiation*

Definitions
(NSA Glossary of Terms *)

- **Availability** – Assuring information and communications services will be ready for use when expected.
- **Integrity** – Assuring information will not be accidentally or maliciously altered or destroyed.
- **Confidentiality** – Assuring information will be kept secret, with access limited to appropriate persons.
- **Authentication** – To positively verify the identity of a user, device, or other entity in a computer system, often as a prerequisite to allowing access to resources in a system.
- **Nonrepudiation** – Method by which the sender of data is provided with proof of delivery and the recipient is assured of the sender's identity, so that neither can later deny having processed the data.

* http://www.sans.org/newlook/resources/glossary.htm; cf. *Trust in Cyberspace*, National Research Council, 1999, Glossary

Developing a Characterization under the Framework: Vulnerabilities/Attacks

- 2. The system or technology may not be able to achieve its functional goals because of certain vulnerabilities* or attacks* (or threats*)

Definitions
(NSA Glossary of Terms *)

- Vulnerability – Hardware, firmware, or software flow that leaves an automated information system (AIS) open for potential exploitation. A weakness in automated system security procedures, administrative controls, physical layout, internal controls, and so forth, that could be exploited by a threat to gain unauthorized access to information or disrupt critical processing.
- Attack – An attempt to bypass security controls on a computer. The attack may alter, release, or deny data. Whether an attack will succeed depends on the vulnerability of the computer system and the effectiveness of existing countermeasures.
- Threat – The means through which the ability or intent of a threat agent to adversely affect an automated system, facility, or operation can be manifest. A potential violation of security.

* http://www.sans.org/newlook/resources/glossary.htm; cf. *Trust in Cyberspace*, National Research Council, 1999, Glossary

Developing a Characterization under the Framework: Countermeasures

- 3. However, the system or technology may counter the vulnerabilities or attacks by protection mechanisms/means that are intended to provide for its particular attributes and assure that it achieves its functional goals

TAV Taxonomies

- **Vulnerabilities, attacks, and threats**
  - Have been considered according to various taxonomies
    - Landwehr, C. E., Bull, A. R., McDermott, J. P., Choi, W. S., "A Taxonomy of Computer Program Security Flaws." *ACM Computing Surveys*, 26(3), September 1994
    - Krsul's Thesis at https://www.cerias.purdue.edu/techreports-ssl/public/97-05.pdf
    - Howard's Thesis at http://www.cert.org/research/JHThesis/Word6/
    - Lough's Thesis at http://scholar.lib.vt.edu/theses/available/etd-04252001-234145/
  - Have been enumerated in databases
    - Common Vulnerabilities and Exposures at http://cve.mitre.org/
    - ICAT Metabase at http://icat.nist.gov/icat.cfm
    - CERIAS Cooperative Vulnerability Database at https://coopvdb.cerias.purdue.edu/main/index.html

Partial Taxonomy

- **Vulnerabilities and attacks**
  - Form a very large class, potentially infinite, which is growing daily
  - Can be viewed according to *when* they arise: at design phase, at implementation phase, or at operational phase
  - May be considered according to *where* they impair a system, *how* they impair a system, or *what* they impair in a system

Outline of a Characterization

1. Technology Description and Information Assurance/Survivability Problem Addressed
2. Assumptions
3. Attacks and Vulnerabilities
4. Information Assurance and Survivability Attributes/Security & Survivability Goals
5. Comparison with Other Systems (Optional)
6. Information Assurance and Survivability Mechanisms
7. Rationale
8. Residual Risks, Limitations, and Caveats
9. Cost and Benefit Analysis
10. References

Problem Addressed

- **1. Technology Description and Information Assurance/Survivability Problem Addressed**
  - What functionality is the technology trying to provide and what in brief are its information assurance and survivability objectives? What is its domain of application?
  - Aims to provide a brief high-level description of functionality and information assurance and survivability objectives
  - Should provide the domain of application and explain limitations
  - Can be extracted from project information or documentation: PI briefings, papers, documents, discussions with PI

Assumptions

- 2. Assumptions
  - What are the assumptions upon which the technology depends?
  - Other technologies may be assumed as supporting the system or technology being characterized
  - Can be divided into assumptions about system, user, network, environment, other technologies
  - May include working hypotheses as special assumptions
  - Provided in the project literature or from PI

Proof Carrying Code: Assumptions

A1	The specification of the instruction set in the logic framework correctly matches the actual behavior of the underlying hardware (manufacturer correctly implements specification, no memory bit-errors, no attacks by voltage variation, etc.).
A2	Capability management: host's access control policy, written by host administrator in our expressive policy language, is appropriate to host's needs.
A3	Digital signatures are only generated by holder of private key; private key is always dept private. *This is only used by the Proof-Carrying Authentication Work.*

Threats, Attacks & Vulnerabilities

- 3. Vulnerabilities and Attacks
  - What are the vulnerabilities and/or attacks that the technology is trying to address?
  - Defined to include any circumstances with potential harm to the system in the form of destruction, disclosure, adverse data modification, and/or denial of service
  - Can be grouped systematically according to design, implementation, and operation (*when* the vulnerability or attack may have its effect)
  - Provided in the project literature or from PI

PCC: Threats, Attacks, Vulnerabilities

Design	
TAV-1.1	Exploitable inconsistency in policy
TAV-1.2	Erroneous decision procedure for granting access or running untrusted program
Implementation	
TAV-2.1	Bug in implementation of protection mechanisms
TAV-2.2	Bug in implementation of decision procedure
Operation	
TAV-3.1	Client code dereferences address outside its own space
TAV-3.2	Client code jumps to address outside itself that's not an API entry point (bypassing access controls)
TAV-3.3	Inconsistency in link-loading name resolution
TAV-3.4	Client code doesn't execute what's checked
TAV-3.5	Forging of certificates
TAV-3.6	Attacker uses compromised keys

IA&S Attributes/ Security & Survivability Goals

- 4. Information Assurance and Survivability Attributes
  - What attributes among system availability (AV), integrity (I), confidentiality (C), authentication (AU), and nonrepudiation (NR) is the technology trying to support?

PCC: Attributes Addressed

		AV	I	C	AU	NR	F
Design	TAV-1.1		□	□	□		□
	TAV-1.2		□	□	□		
Implementation	TAV-2.1		□	□	□		□
	TAV-2.2		□	□	□		
Operation	TAV-3.1		□	□			
	TAV-3.2		□	□			
	TAV-3.3		□	□			
	TAV-3.4		□	□			
	TAV-3.5				□		
	TAV-3.6				□		

Mechanisms

- **6. Information Assurance and Survivability Mechanisms**
  - What techniques are used to mitigate given vulnerabilities and attacks? Examples are:
    - Damage assessment
    - Containment
    - Reconfiguration
    - Repair
    - Fault treatment
  - Intended as support for the high-level information assurance and survivability attributes

PCC: Mechanisms

M1	Prover: constructs safety proof for untrusted application binary [Nec97]
M2	Machine specifications: axiomatizes behavior of machine instructions [MA00]
M3	Safety policy: defines "theorem" to be proved [App01]
M4	Proof checker: determines whether proof matches theorem [AMSV02, PS99]
M5	Policy Modeler: validation technique for safety policies [AF01]
M6	Semantics of types: safety proofs for advanced type systems [AF00]
M7	Use of digital signatures (can be generated only by holder of private key)
M8	Expiration: "freshness dating" of certificates helps limit damage from key leakage
M9	Type-safe linking and position-independent code [CWAF02]

Rationale

- **7. Rationale**
  - How do the elements fit together? Provide a rationale matrix
  - Footnote for each mechanism/assumption cell of the matrix
    - Descriptive paragraph showing that the assumptions and mechanisms counter the vulnerabilities and attacks and thus supporting claims about achieving the high-level attributes
  - N.B.: *Rationale matrix plus footnotes only outline the beginning of validation*; a validation plan is needed; validation comes afterwards and is likely to involve significant additional effort

Validation Techniques

- Techniques for verification and validation include
  - Red team testing and analysis
  - Formal assurance argument
  - Formal methods of proof
  - Modeling and simulation
  - Code inspection
  - Cryptanalysis
  - Other techniques
- Independent peer review
- Summary

PCC: Rationale

		AV	I	C	AU	NR	F
Design	TAV-1.1		A2, M5[1]				M1, M3, M6[2]
	TAV-1.2		M4[3]				
Implementation	TAV-2.1		TCB[4]				M1, M3, M6[2]
	TAV-2.2		M4[3]		M4[8]		
Operation	TAV-3.1		M2, M3, M4[5]				
	TAV-3.2						
	TAV-3.3		M9, note[6]				
	TAV-3.4		M2, M3, M4[7]				
	TAV-3.5				A3, M7, note[9]		
	TAV-3.6				M8, note[9]		

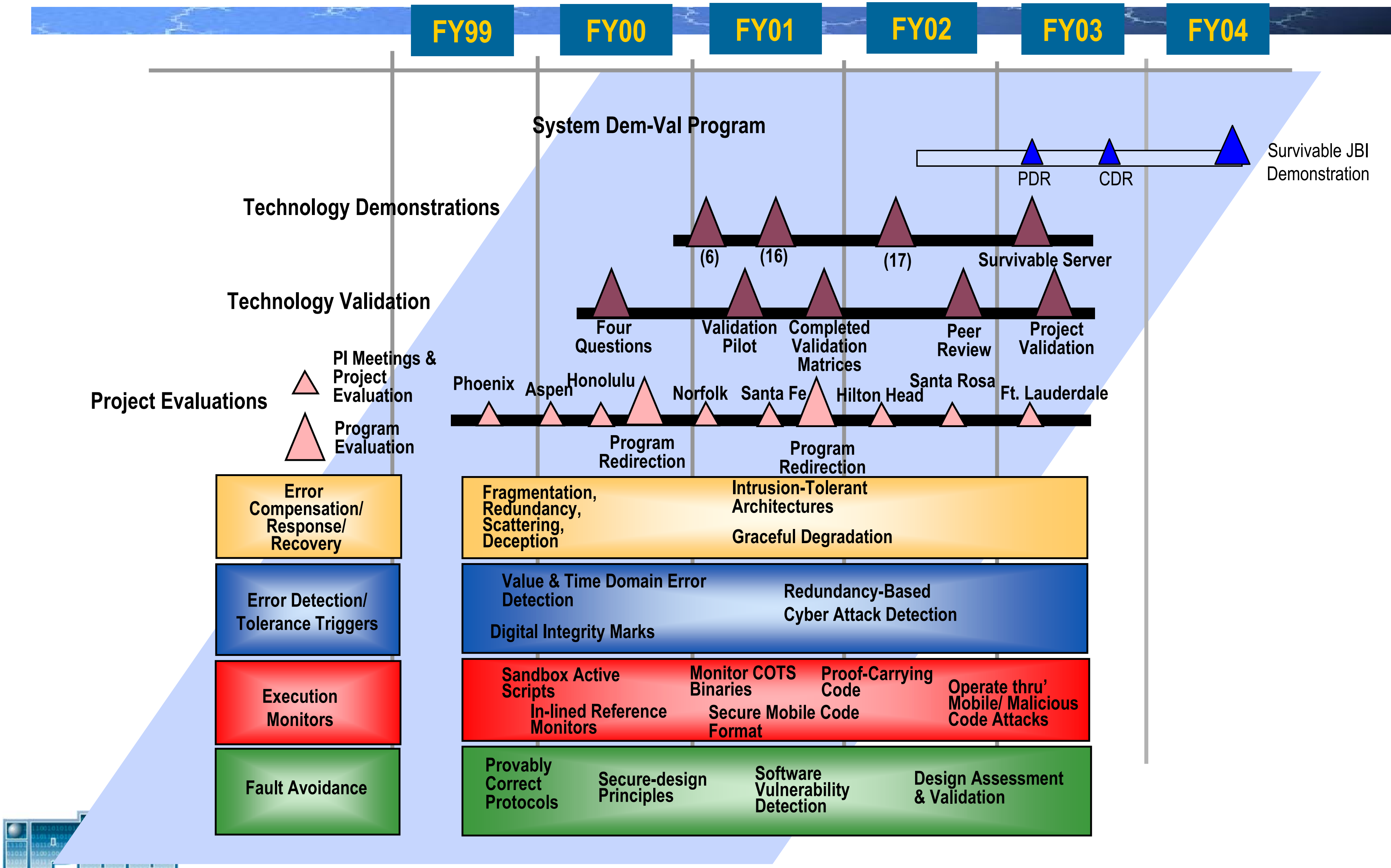
Residual Risks

- **8. Residual Risks, Limitations, and Caveats**
  - What are the residual risks or gaps?
  - Residual risks may relate to other technologies assumed to support the system or technology being characterized
  - These may be determined from the arguments under the rationale in 7

Costs

- **9. Cost and Benefit Analysis**
  - What are the costs with respect to the benefits?
  - Cost metrics (quantified if possible)
    - Performance degradation
    - Functionality change
    - Storage needs
    - Network bandwidth requirements
    - Cost as $
  - Benefit metrics (quantified if possible)
    - Probability of surviving an attack, loss of data, loss of confidentiality
    - Length of time in successfully defending against attacker
  - One-to-one correspondence of mechanisms to goals

OASIS Roadmap



FY99 FY00 FY01 FY02 FY03 FY04

System Dem-Val Program

PDR CDR

Survivable JBI Demonstration

Technology Demonstrations

(6) (16) (17) Survivable Server

Technology Validation

Four Questions

Validation Pilot

Completed Validation Matrices

Peer Review

Project Validation

PI Meetings & Project Evaluation

Program Evaluation

Project Evaluations

Phoenix Aspen Honolulu Norfolk Santa Fe Hilton Head Santa Rosa Ft. Lauderdale

Program Redirection

Program Redirection

Error Compensation/ Response/ Recovery

Fragmentation, Redundancy, Scattering, Deception

Intrusion-Tolerant Architectures

Graceful Degradation

Error Detection/ Tolerance Triggers

Value & Time Domain Error Detection

Digital Integrity Marks

Redundancy-Based Cyber Attack Detection

Execution Monitors

Sandbox Active Scripts

In-lined Reference Monitors

Monitor COTS Binaries

Secure Mobile Code Format

Proof-Carrying Code

Operate thru' Mobile/ Malicious Code Attacks

Fault Avoidance

Provably Correct Protocols

Secure-design Principles

Software Vulnerability Detection

Design Assessment & Validation

OASIS Program: Validation Dimensions

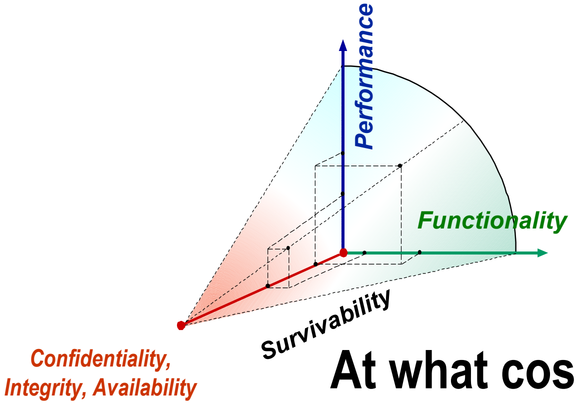
	Integrity	Availability	Confidentiality	Authentication	Non-repudiation
ILoveYou	🟩	🟩	🟩	N/A	N/A
Anna Kournikova	🟩	🟩	🟩	N/A	N/A
Nimda	🟩	🟩	🟩	N/A	N/A
Code Red I & II	🟩	🟩	N/A	N/A	N/A
Stachaldracht	🟩	🟩	N/A	N/A	N/A

Is intrusion tolerance feasible? - *Yes*

OASIS Program: Validation Dimensions

	Integrity	Availability	Confidentiality	Authentication	Non-repudiation
ILoveYou	🟩	🟩	🟩	N/A	N/A
Anna Kournikova	🟩	🟩	🟩	N/A	N/A
Nimda	🟩	🟩	🟩	N/A	N/A
Code Red I & II	🟩	🟩	N/A	N/A	N/A
Stachaldracht	🟩	🟩	N/A	N/A	N/A

Is intrusion tolerance feasible? - *Yes*



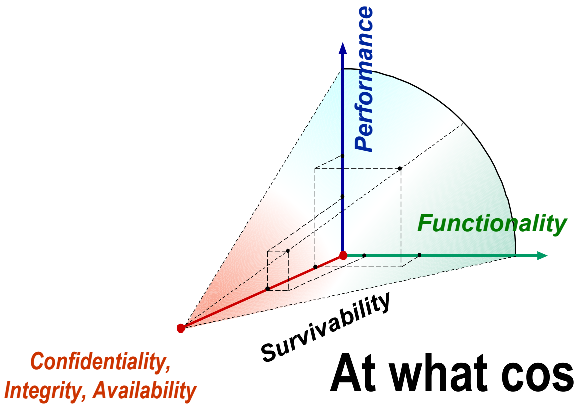
At what cost?

- Performance Overheads Quantified

OASIS Program: Validation Dimensions

	Integrity	Availability	Confidentiality	Authentication	Non-repudiation
ILoveYou	🟩	🟩	🟩	N/A	N/A
Anna Kournikova	🟩	🟩	🟩	N/A	N/A
Nimda	🟩	🟩	🟩	N/A	N/A
Code Red I & II	🟩	🟩	N/A	N/A	N/A
Stachaldracht	🟩	🟩	N/A	N/A	N/A

Is intrusion tolerance feasible? - *Yes*



At what cost?

- Performance Overheads Quantified

Proof-Carrying Code Project	Integrity	Availability	Confidentiality	Authentication	Non-repudiation
Policy inconsistency		🟩	🟩	🟩	
Decision procedure		🟩	🟩	🟩	
Bug in protect. mech.		🟩	🟩	🟩	
Bug in decision proc.		🟩	🟩	🟩	
Illegal fetch/store		🟩	🟩		
Illegal jump		🟩	🟩		
Name resolution		🟩	🟩		
Check A, Execute B		🟩	🟩	🟩	
Forge certificate				🟩	
Compromised keys				🟩	
Unauthorized delete					
Invalid permissions					

Which security attributes are assured? Against which attacks/vulnerabilities?

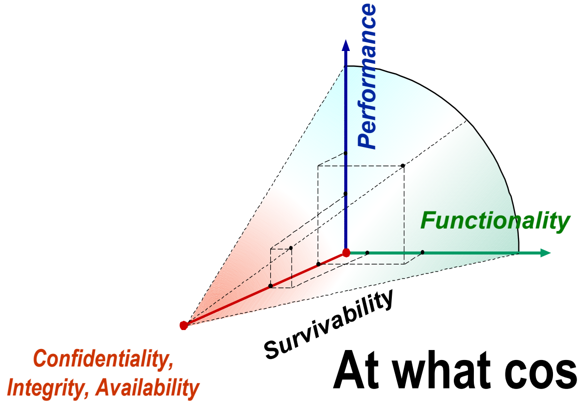
OASIS Program: Validation Dimensions

	Integrity	Availability	Confidentiality	Authentication	Non-repudiation
ILoveYou	🟩	🟩	🟩	N/A	N/A
Anna Kournikova	🟩	🟩	🟩	N/A	N/A
Nimda	🟩	🟩	🟩	N/A	N/A
Code Red I & II	🟩	🟩	N/A	N/A	N/A
Stachaldracht	🟩	🟩	N/A	N/A	N/A

Is intrusion tolerance feasible? - *Yes*

Proof-Carrying Code Project	Integrity	Availability	Confidentiality	Authentication	Non-repudiation
Policy inconsistency		🟩	🟩	🟩	
Decision procedure		🟩	🟩	🟩	
Bug in protect. mech.		🟩	🟩	🟩	
Bug in decision proc.		🟩	🟩	🟩	
Illegal fetch/store		🟩	🟩		
Illegal jump		🟩	🟩		
Name resolution		🟩	🟩		
Check A, Execute B		🟩	🟩	🟩	
Forge certificate				🟩	
Compromised keys				🟩	
Unauthorized delete					
Invalid permissions					

Which security attributes are assured? Against which attacks/vulnerabilities?



At what cost?

- Performance Overheads Quantified

OASIS Program	Integrity	Availability	Confidentiality	Authentication	Non-repudiation
Malicious Code					
DOS					
Insider Attack					

Coverage?

Validation: Future Research Areas

- Concepts and terminologies to succinctly express IA domain issues
- Threat, attack and vulnerability taxonomies
- Security models and models of attacker intent, objectives, and strategies
- Work factor metrics, survivability metrics, operational security metrics, cryptographic protocol metrics
- Methods for testing and validating protection mechanisms
- Security and survivability requirements specifications