Software Application Security

Martin R Stytz, Ph.D.
Jeff Hughes
AFRL/SNAS
2241 Avionics Circle
WPAFB, OH 45433-7320

SPI - Mission

- Software Protection Initiative (SPI)
  - To prevent the unauthorized distribution and exploitation of national security applications by our adversaries

SPI - Motivation

- High performance computing (HPC) hardware availability (i.e., Linux clusters)
- Decades of investment in high performance software and the research results they embody
- Critical to every aspect of military activity, from training to operations
- Protected software is the foundation for high confidence computing

SPI - Vision

- Establish the Software Protection Initiative as an integral layer of the defense in depth concept for information assurance
- Complement existing information assurance efforts in network security and operating system access controls with an application-centric approach to protecting critical DoD intellectual property

SPI - Strategy

- Develop technologies that provide quantifiable protection of sensitive software
- Determine resilience of technologies to attack or subversion
- Sub-components
  - Continual assessment of technologies
  - Development of techniques and technologies
  - Development of metrics and benchmarks

Application Security Definition

What is meant by "Application Security"

- Anti-Piracy
  - Protected distribution
  - Protected execution
- Code integrity
  - Trusted execution
- Vulnerability reduction
  - Reduction of security flaws
  - Secure development environment

SPI - Scope

- What SPI is: Securing high value application software running on COTS computers.
  - Presently consists of 3 thrusts:
    - Identify and protect existing critical applications
    - Devise secure development environment for future applications
    - Educate the DoD community
- What SPI is not:
  - Network security
  - Operating system access control

SPI - Goals

- Institutionalize software protection as part of the application software life-cycle
- Educate and train the community
- Develop a wide array of user-friendly protection techniques
- Ensure that protection technology and policy are appropriately applied to protect and extend our technological advantage

SPI Activities

- Training
- Protection Technology VV&A
- Metrics Development
- Outreach & Education
- Research & Development

SPI Activity Training

- Goal
  - Train program managers, engineers, scientists, and software developers on how to protect code pedigrees and how to write protectable code
- Key Activities
  - Developing modular short course
    - Formal training courses will be held at SPC and other locations approximately six times per year

SPI Activity
Protection Technology VV&A

- Goal
  - Respond to user/developer feedback and validate usability, scalability, and maintainability of SPI technologies
- Key Activities
  - Assembled broad based VV&A support structure
    - Technology Review Panel
    - Internal Red Team, VV&A
    - External Red Team
    - Insertion Team
    - User Community

SPI Activity Metrics

- Metric Categories
  - Denial of use
  - Denial of exploitation
  - Validate “ilities”: usability, scalability, maintainability, availability
- Values considered for criteria

Cost to “us”	vs.	Cost to “them”
$$ to implement		**$$ to defeat**
Run-time impacts		**Time to defeat**
Skills needed to implement		**Skills needed to defeat**
Extra memory usage		**Size of team needed**
Numerical accuracy		
Schedule impact		
Reliability		

SPI Activity
Outreach & Education

- Goal
  - Cultivate awareness of the threat and the need for application code security
  - Promote software protection as integral to defense in depth for Information Assurance.
- Key Activities
  - Academic centers of excellence program
  - SPI is participating in conferences, providing input for publications, and establishing contacts to increase awareness of the need for software protection

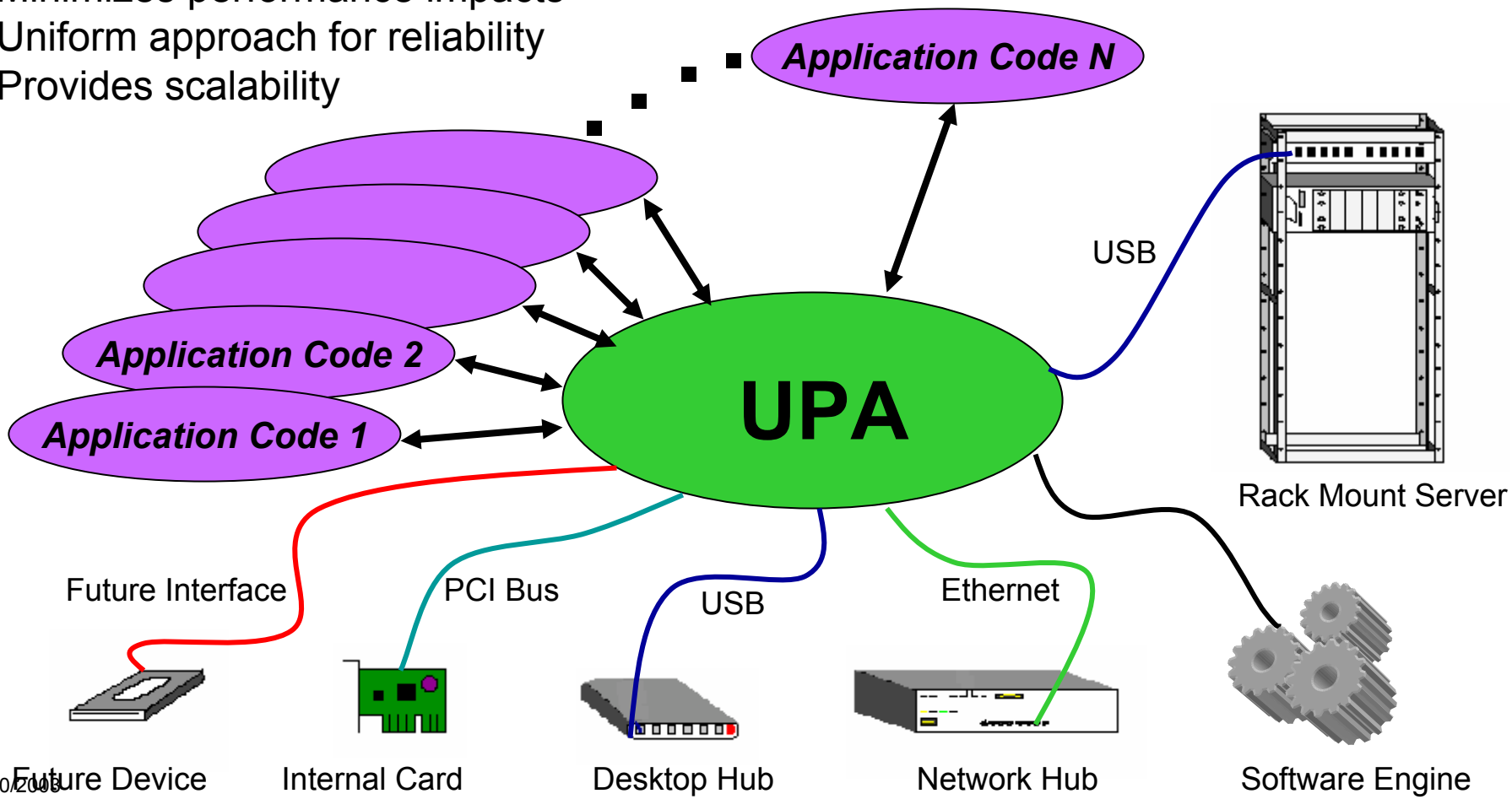
SPI Activity
Research & Development

- Goal
  - Advance software protection technologies on desktops through super computers
- Focus
  - protect developed software
  - Develop protectable software
- Key Activities
  - Protect developed code
  - Develop protectable code
  - Promote usability, scalability, and maintainability
  - Universal Protection Architecture

R&D Activity

Unified Protection Architecture (UPA)

- Minimizes performance impacts
- Uniform approach for reliability
- Provides scalability

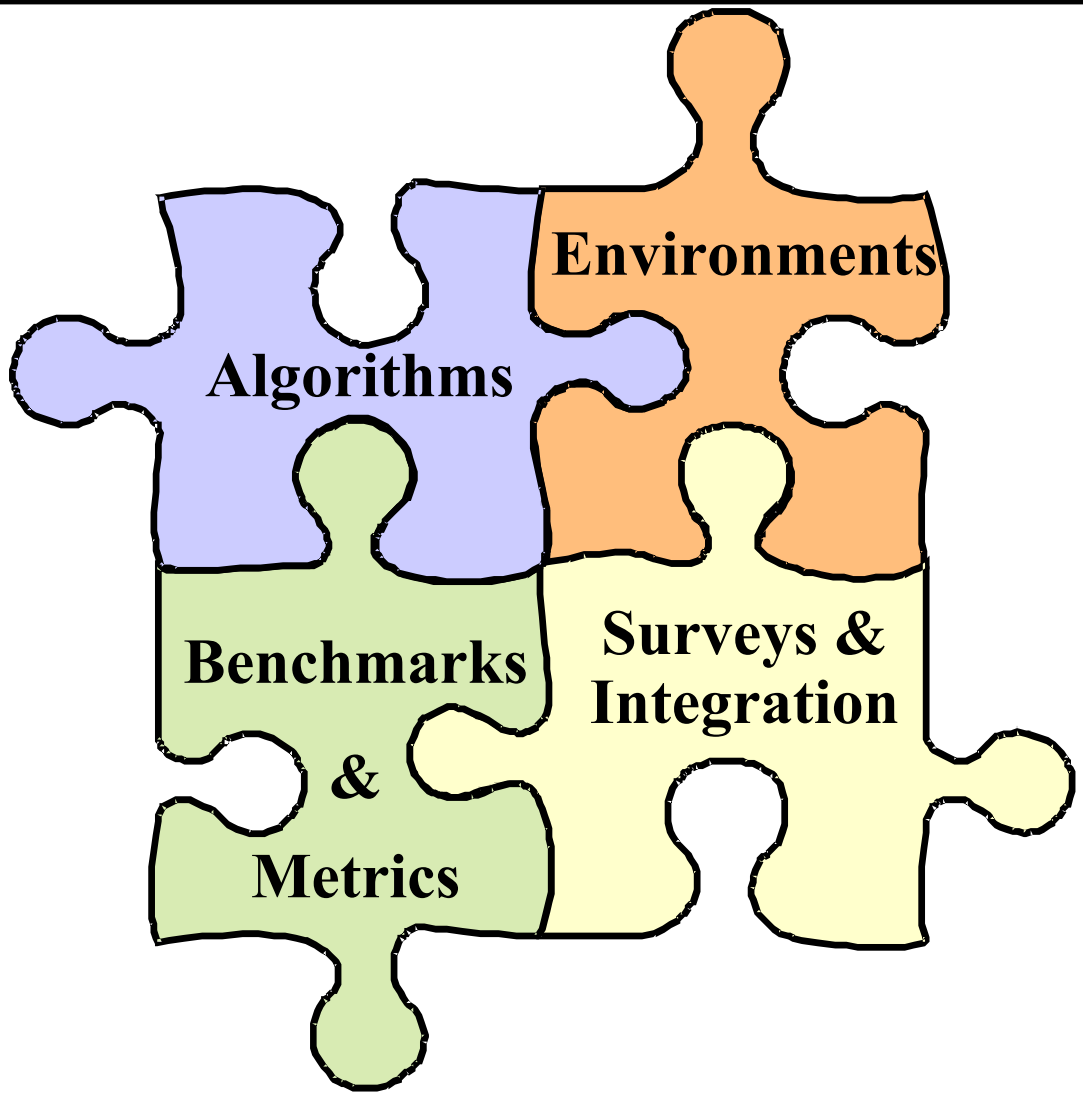


R&D Strategy Vision*

- Protection
  - Application security without development or performance penalty
  - Protection techniques tailored to the criticality of the code, the operational and threat environments, and computational power
  - Scalable and customizable protection
- Detection
  - Self monitoring of protected software for
    - Malicious activity
    - Code integrity
- Reaction
  - Array of autonomous self defense measures for protected codes
    - Modification of code/data
    - Self destruction
    - Reporting

**Secrets and Lies: Digital Security in a Networked World*, Bruce Schneier, John Wiley and Sons, Inc., 2000

Research Areas



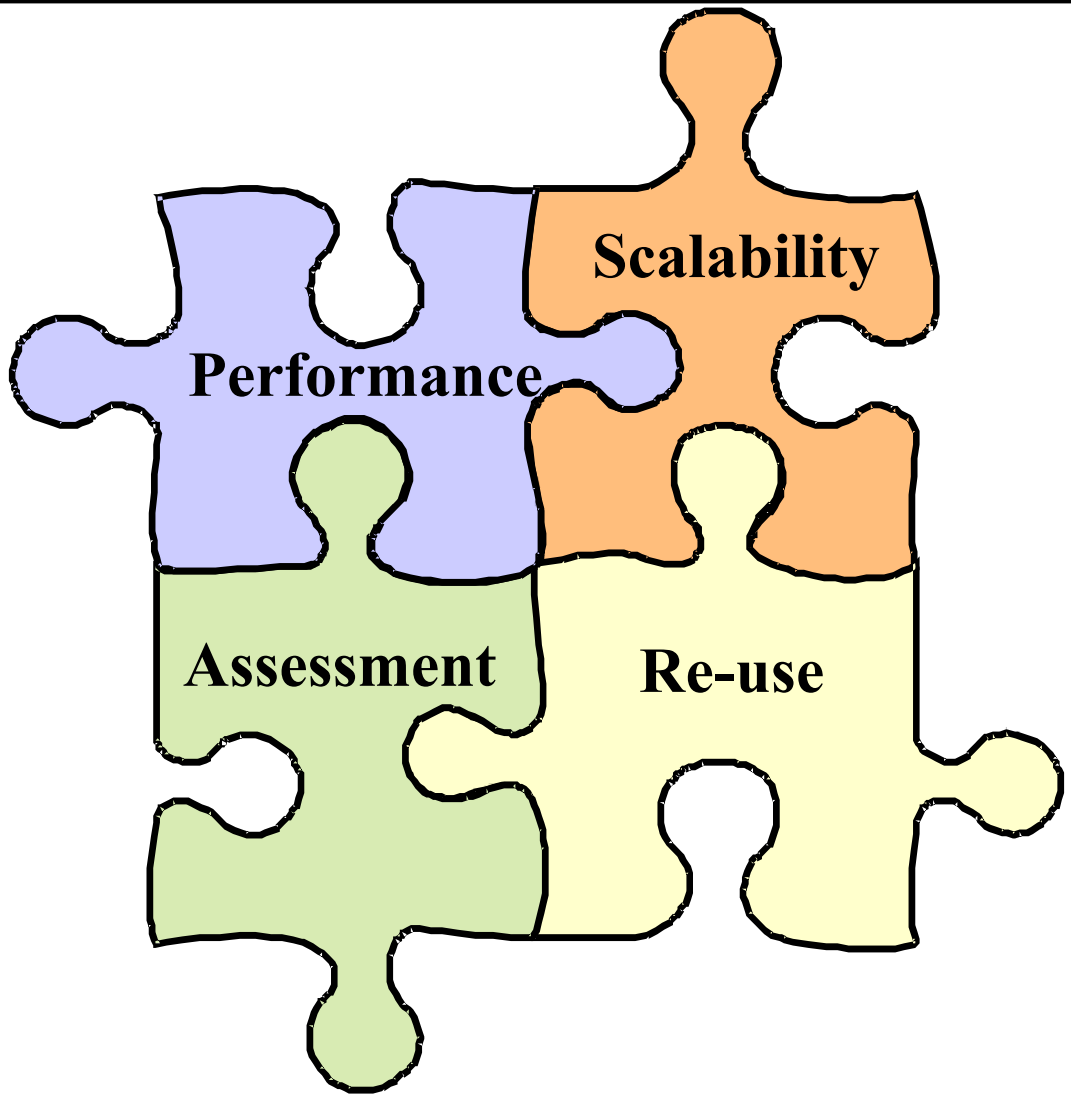
Research and Development Dimension of the Challenge

- Framework for Analysis
- Technology for achieving SPI goals
- Quantification of code complexity
- Performance metrics

Framework for Analysis

- Goals of attacks
  - Reverse engineering all or parts of a code
  - Allowing limited or unrestricted execution
  - Tampering with the code
- Type of effort
  - Human effort (from expert to ordinary skills)
  - Generic tool availability (COTS, open source)
  - Specialized tools (what is possible by skilled adversaries?)
  - Number of allowed executions
  - Time and availability of code required for attack
  - Level of mathematical or logical symbolic analysis

Research Issues



R&D Technology Hardware Approaches

- Freestanding, obfuscated code only
- Obfuscated code with an “authentication” host on the network
  - Kind of network
  - Kind of host processing
- Obfuscated code with on-board hardware module/hardware
  - Proprietary
  - TCPA, Palladium, COTS
- Other approaches and combinations of the above

R&D Technology
Software Approaches

- Obfuscation of code
  - At source code level
    - source restructuring
  - At executable level
    - Obfuscating compiler
    - Post-compilation obfuscation
  - Three address code representation

- Opaqueness of code and/or procedures
  - Obfuscate procedures
  - Complexity of parameters passed, nesting
  - Use of special hardware for function evaluations

Quantification of Code Complexity

- At source, intermediate and/or executable levels
- Basic block count and size
- Structure of program control flow graph
  - At basic block level
  - Static analysis
  - Dynamic analysis for a “typical” execution
  - Loop structure and depth
- Data structure complexity
- Procedure call depth, count, parameter passing (indirection, etc.)

R&D Metrics

- Performance metrics
  - Possible levels of protection as a function of code complexity and attack effort
  - Performance loss as a function of protection
  - Preprocessing effort required for protection
  - Cost of protection – hardware, management, etc
  - Cost of versioning, updating, etc.

Research Objectives

- Protect developed code
- Develop protectable code

Protecting Developed Code

- Continually assess the state-of-the-art
  - Develop capabilities to maintain security edge in face of technological advances
- Areas of concern/research interest
  - Decompilers
  - Watermarking
  - Compilers
  - Multiprocessors
  - Disassemblers
  - Obfuscation
  - Debuggers
  - High Performance Computing

Developing Protectable Code

- Continually assess state-of-the-art
- Areas of concern/research interest
  - Secure development environments
  - Automatic pedigree generation and validation
  - Automatic developer logging and profiling
  - Software development methodology modification
  - Virtual machine wrappers
  - Multiprocessors

R&D
Current Efforts

- Development of topics
  - Obfuscation and Watermarking
  - Tampering & Reverse Engineering
  - Architectural Degradation
  - Tamper Detection & Response
  - Binary Code Transformation
  - AT Protection Thru Obfuscation

R&D
Protection Research Avenues

- Benchmarks, metrics, and test suites
  - Autonomous red team
  - Ontology and lexicon
- Secure development environment
  - Architecture through maintenance phases
- Black box application of protection technologies
- Cross authentication of components
- Improved watermarking and obfuscation

R&D Strategy
Protection Research Avenues (cont)

- Autonomous, secure assembly and verification of security capabilities
  - Composable protection techniques
- Data
  - Container-based protection of data
- Inherently secure programming languages
- Multiprocessor software protection
- Operation on untrusted hardware

R&D Strategy
Detection Research Avenues

- Autonomous attack detection and defense
- Ontology and lexicon
- Comprehensive threat description and threat models
- Voting schemes to “detect” subverted software or nodes
- Continuous or pushbutton verification that the software is not changed
- Security gauges

R&D Strategy
Reaction Research Avenues

- Adaptive defense
- Variable precision and accuracy
- Benchmarks and test suites
- Autonomous recovery and repair
- Isolation of subverted nodes
- Secure migration of subverted processes
- Pedigree to track back to developer

Strategic Issues

- Education
- Technical Thrusts
- Government-wide Coordination
- Risk Management

Strategic Issues
Education

- Issue
  - Education is critical to cultivate awareness of the threat and the requirement for application code security across the DoD
- Discussion
  - Education is required at all levels
  - SLAG and IPT must have reps from key DoD and government agencies and assist in education process
  - SPI will encourage commercial entities to issue statements supporting the initiative
  - Web will be a key education tool
- Way Ahead
  - Involvement is essential…..

Strategic Issues
Technical Thrusts

- Issue
  - SPI must identify and protect existing critical codes and develop secure software development tools/environments for future applications
- Discussion
  - Currently rely on the inherent obfuscation provided by current higher order language compilers
- Observation
  - Ensure R&D investments address these core issues and backstop the technology risk
  - Developing comprehensive and integrated R&D strategy to meet short and long term objectives

Strategic Issues
Government-Wide Coordination

- Issue
  - Critical applications are shared among government organizations
  - Software protection policy, techniques, and procedures must be consistent
- Discussion
  - All actions must be coordinated at senior levels
    - National Cyberspace Policy
    - DoE, NASA, and others
      - Sharing of applications and procedures
      - Common requirements definition
- Way Ahead
  - Include key government organizations in activities

Strategic Issues
Risk Management

- Issue
  - SPI program success requires balancing multiple, competing factors
- Discussion

Established DoD acquisition process	**vs.**	**Typical academic-based software development process**
Compartmentalization of facts	**vs.**	**Education**
Strong protection measures	**vs.**	**Ease of use**
Directed compliance	**vs.**	**Voluntary implementation**
DoD only	**vs.**	**Government-wide implementation**

- Observation
  - R&D to enable usability, scalability, and maintainability
  - Definitive policy to institutionalize SPI
  - Education and coordination to encourage compliance

Conclusion

- Application software represents a significant portion of the DoD's intellectual property
  - Significant investment in both time and money
  - Enables development of next generation weapon systems
- Protecting critical application software allows U.S. Forces to:
  - Maintain a technological advantage over our adversaries
  - Extend the operational life of critical systems

Conclusion, cont.

- Software protection is an integral layer of the defense in depth concept for information assurance
  - Compliments network security and access controls
  - Provides application centric technology to reinforce application security policy