Summary: Now

- Choices seem to be poor quality COTS or tired MLS-inspired systems. Neither is acceptable. We need to re-examine fundamental qualities.
- Cost of acquisition drives too many decisions. TCO, help-desk and add-ons should also be considered.
Bugs not a new problem:

• Preliminary Notes on the Design of Secure Military Computer Systems, Roger Schell, USAF
  January 1, 1973
  “…From a practical standpoint the security problem will remain as long as manufacturers remain committed
to current system architectures, produced without a firm requirement for security. As long as there is
support for ad hoc fixes and security packages for these inadequate designs and as long as the illusory results of penetration teams are accepted as demonstrations of a computer system security, proper security will not be a reality.”
Areas for Research/Deployment
Policy

• How do we
  – formally express policy?
  – represent policy in a system?
  – support dynamic policy (coalitions)?

• How can we
  – model and evaluate policies?
  – automatically enforce policy?

• Policy traceable to features
• Natural language expression
• Policy “libraries”
Construction

• How can we build secure software
  – At reasonable cost
  – Quickly
  – That is testable
  – That interoperates without danger

• How do we test security?
Reliable Metrics

• How secure is my system?
• How secure is my network?
• Is a change worthwhile?
• What is the affect of adding new exposure?
• How can I balance protection level with cost?
• Which architecture will be safer?
Assured Availability

- Resistance to attack
- Resistance to failure
- Automatic reconfiguration & recovery
- Graceful degradation under attack
- Formal models of availability and Quality of Service (QoS)
Accurate Risk Data

- How likely is a threat?
- How likely is an attack?
- How likely is a failure?
- What is the likely loss?
- Provide feedback to policy decisions
- How to collect, classify, and organize appropriately?
- Dynamic evaluation
Graceful Penetration Tolerance

• Attacked but contained
• Automatic reconfiguration
• Fallback configurations and systems
• Automatic deployment of recovery mechanisms
• Non-stop operation. “Survivability” in DARPA-speak.
Automated Responses

• Respond to attacks
• Don’t respond in error
  – Not really an attack
  – Mistakes about origins and mechanisms
• Respond only enough to contain or stop?
• Integrate with other systems’ responses
• Also satisfy law-enforcement needs
• Automated “strike-back” is not an option in most cases
Forensics

- Who is coming across the network?
- Where are they coming from?
- Legally-supportable evidence
- What did that software do?
- Who wrote that virus?
- What happened?
- How did it happen?
- Automated analysis of attacks
- Automated distribution of response
Identification and Authorization

- Portable on-line ID
  - Signed biometrics?
- Authorization without Identification
  - Short-term
  - Permanent
- PKI and Public Keys
  - Availability
  - Interrelationship
  - Dynamic keys
  - Revocation management
Useful Audit Trails

• What needs to be logged from
  – Host
  – Applications
  – Network
• How do we store it?
• Dynamic auditing and reconstruction
• Scalability and reduction management
• Real-time access

Current Challenges

Future Challenges
Forensics

- What failed?
- What was read? Altered?
- Who did it?
- Where did it come from?
- What else was involved?
Models

For over a dozen years, we have been focused on the MLS model for secure systems; The “Orange Book” and its progeny.
This model doesn’t really work for networks, object-based systems, thin clients, and active content.

What model should we use?
What alternatives can we create?
How will that model adapt to future architectures?
How can we influence COTS?
Multimedia Security

Systems are processing different kinds of data than text and numbers.
How do we secure systems involving:
– Real time video
– Audio
– Multi-media databases
– Active content
– Remote processing
Database

- Data quality
- Interoperability without disclosure
- Auditing access
- Versioning/labeling
- Derivative and inference control
Human-computer Interface

- Easy to learn
- Easy to use
- Difficult to circumvent
Accountability

• Hold command responsible for poor configuration management and poor choices.
• Hold vendors responsible for poor quality
• Hold operational personnel responsible for sloppy security
Refocus

- Patching to design
- Detection to prevention
- Add-ons to Built-ins
- Reaction to forensics
Future concerns

• More connectivity (“Evertnet”)
• Wireless
• Portability of storage
• Autonomous agents
• Embedded systems
• Greater mixing of personal and task computing
• Dynamic code patching
More concerns

- Strangeness in the law
  - Support for IP
  - Constrained visibility into objects
- Device computing
- Fragility of communications
- EMF/EMP
  - Location finding
- Time synchronization
Auxiliary Concerns

- People, training
- Multinational concerns
- Law enforcement readiness
- Interoperability with civilian computing
- Physical security