SAND2011-1809C

# Defense in Depth

## A Lifecycle and Temporal View

**Raymond C. Parks**

*Principal Member of Technical Staff*

Sandia National Laboratories

Assurance Technologies and Assessments Department

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND2010-0806P

Sandia National Laboratories

# Overview

- Recap
- Security over the lifecycle
- Adversaries and Consequences
- Architecting Security
- Developing Security
- Operating Securely
- Failure to plan for failure
- Retire gracefully

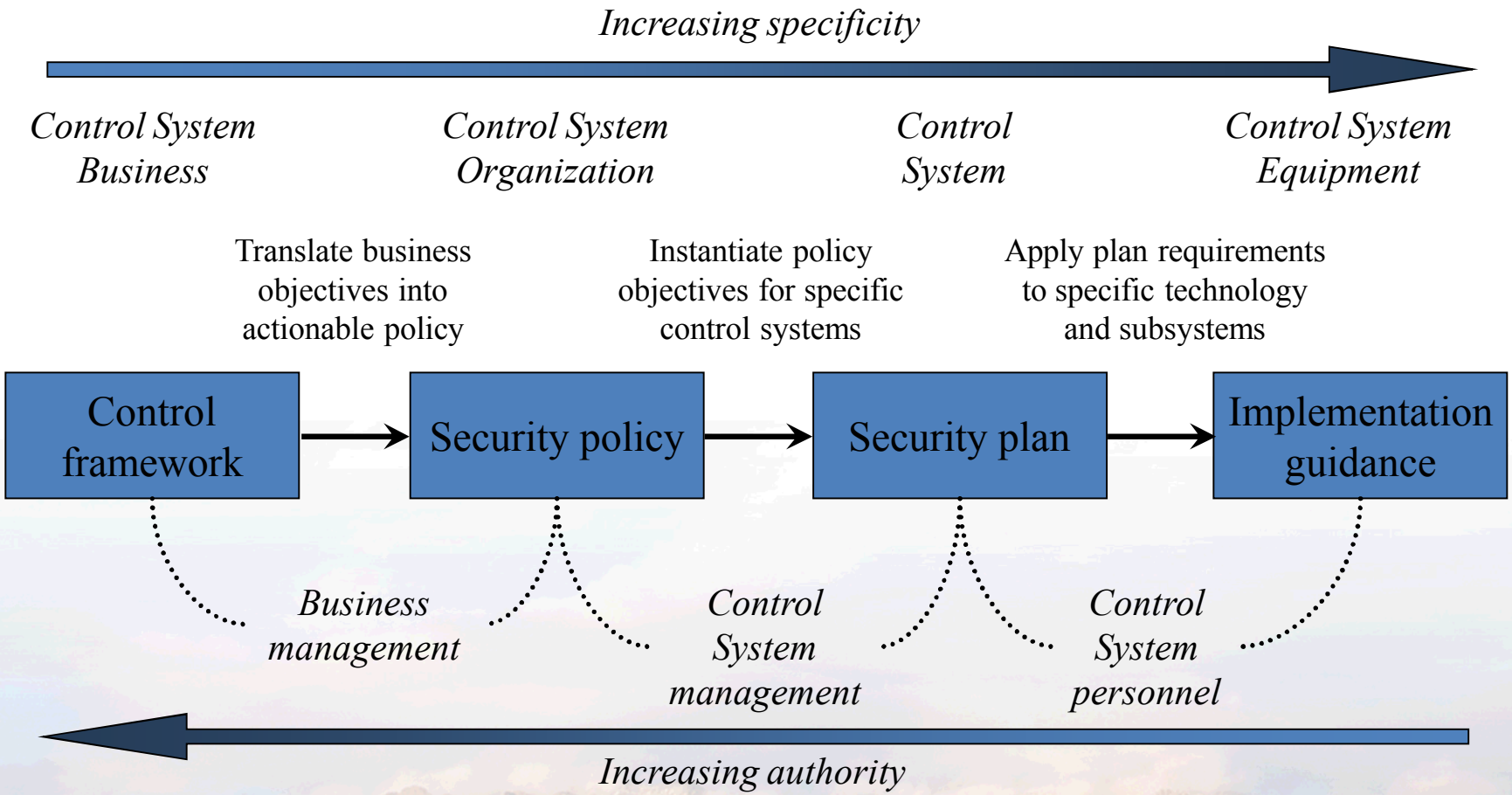
# Three Generations of Computer Security

- 1st Generation - Prevent
  - Develop technologies to prevent any computer security disruption
  - Ended with MULTICS paper in 1974
- 2nd Generation - Prevent and Detect
  - Develop technologies to detect when prevention fails - allows response
  - Ended with “Reflections on Trusting Trust” in 1984
- 3rd Generation - Prevent, Detect and Survive
  - Develop technologies to survive disruptions when prevention and detection fail

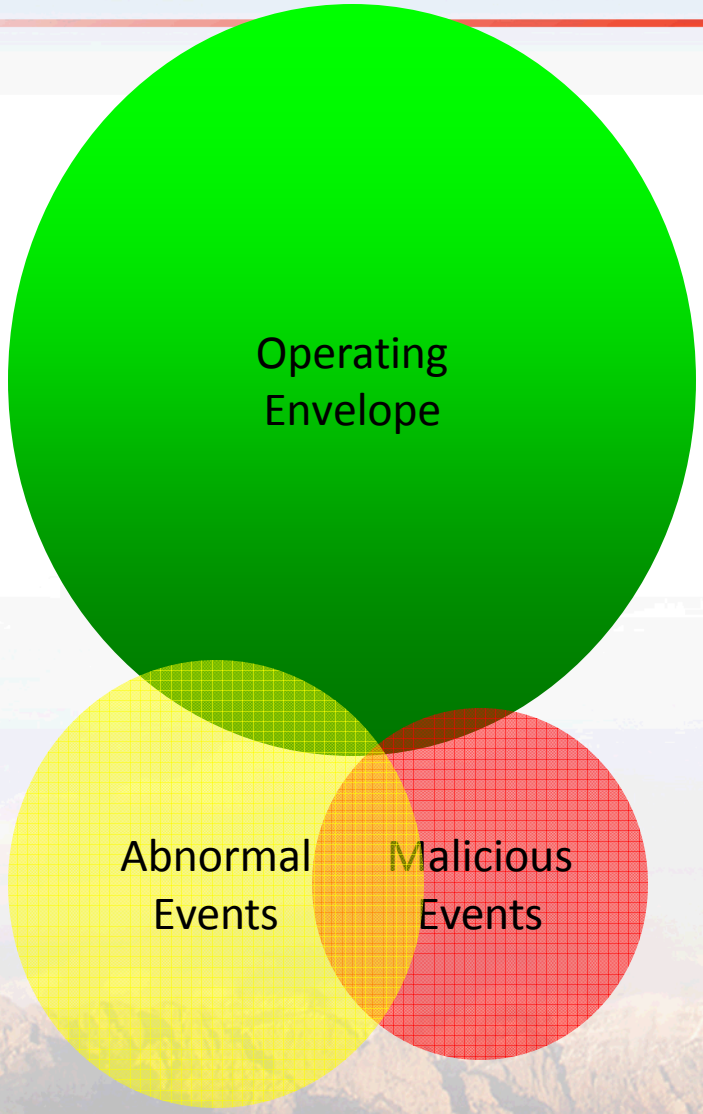
# Security is a System

- Systems include:
  - People,
  - Processes, and
  - Technology
  - to achieve functionality
- Security needs:
  - People,
  - Processes, and
  - Technology
  - To achieve secure function

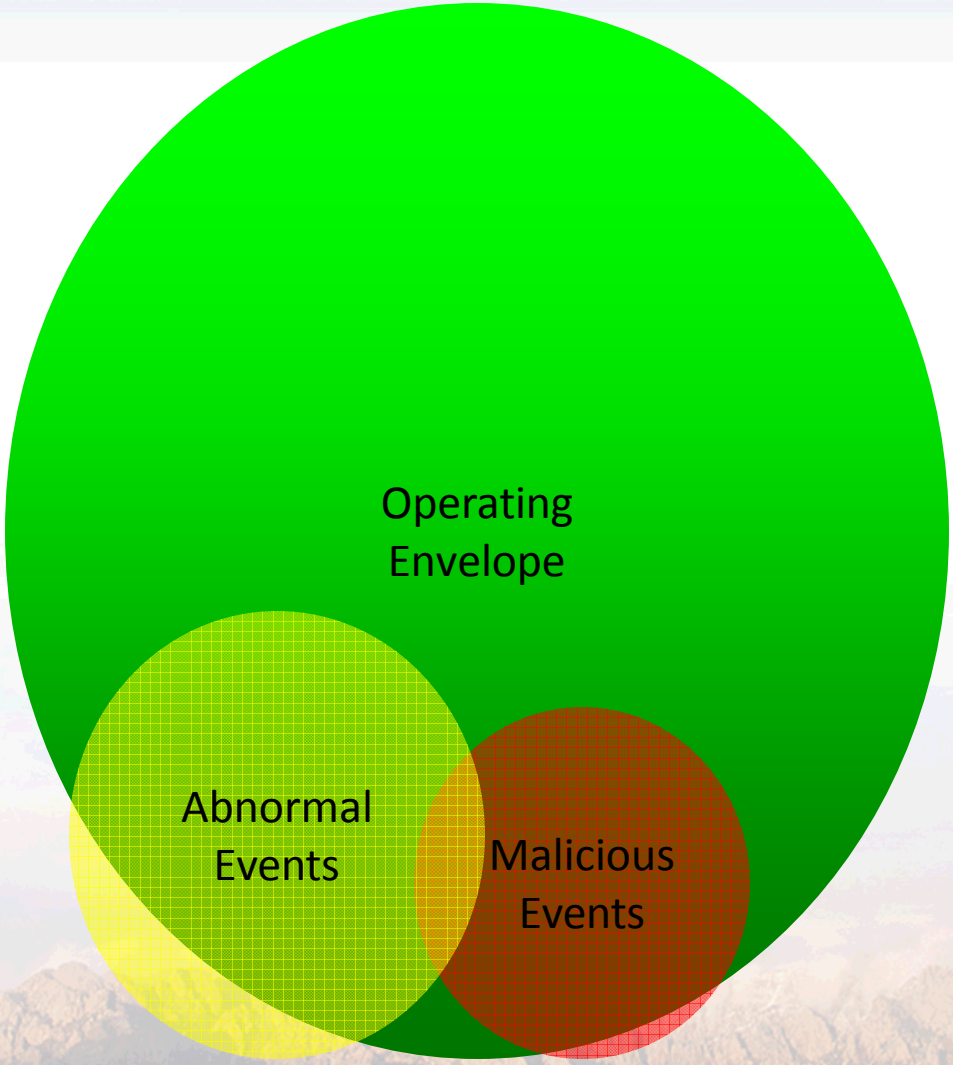
# Sustainable Security



# Current Environment



# Goal Environment



# Security over the lifecycle

- Security duct taped on after deployment will always fail
- Security starts when the system starts
  - System Purpose and Concept of Operations
  - Adversary model and consequences/effects
- Top-level architecture/design can include fundamental security flaws no amount of duct tape will fix
- Development provides opportunity for supply chain attacks on system materials and development systems
- Operations involve implementation failure and more supply chain
- Failure to plan for failure leads to failure
- Don’t give the security away with the retired system

# Adversaries and Consequences

- Any system that does anything non-trivial can be abused to cause consequences
- There will be someone, somewhere who will want to cause those consequences
- Systems support a business/mission purpose
  - Think about what it would mean for that purpose to fail – both in itself and to dependent systems
- Ask yourself – who would want to cause the system to fail at its purpose

# Architecting security

- Architects need to assume that systems in the system will bring security problems along with functional success
- No system exists in a void – trust but verify information and services across connections to other systems
- Keep it simple – don't do more than is necessary
- Plan for the people and processes that will use the technology – architect a system that can only do what you intend
- Think of attacks – and think of solutions

# Developing Security

- Plan to develop security securely – it's a cost of doing business, there's no more ROI than paying for electricity
- Consider what the identified adversary can do to the development effort – advanced adversaries (APT) can do a lot
- It's guaranteed that adversary attack on development can cause the system to fail its purpose
- At least protect the development process and products from attack
- For advanced adversaries, protect the systems you use to develop – software tool chains, malleable hardware design, fixed hardware, mechanical , information

# Operating Securely

- Supply-chain attacks apply just as much as in development – no system operates without maintenance and external contact
- A sound architecture should prevent, detect, and survive misuse cases by operators and maintainers
- Monitor continuously – not just the system but the threats – attacks evolve continuously but your system only evolves when you change it
- Don't give away information unnecessarily – OPSEC – procurement, hiring, and unsatisfied operations personnel needs are an information back-door

# Failure to plan for failure = epic fail

- Assume the worst and have a plan to recover
  - Consider improvements to system – once deployed, people and processes change easier than technology
  - Consider duct tape
  - Consider building recovery capability
  - Maybe you just have to accept the risk – make sure everyone concerned knows and can plan for your system's failure
- Ensure the capability to support mitigation
- Don't get stuck in legacy hell

# Retire gracefully – and securely

- Unless the system is a one-off and has no successor, protect the information about the system when retiring it
- Obvious stuff – shred design and operating documents, wipe electronics
- Non-obvious stuff – protect procurement, staffing, and support information

# Summary

- Security starts when the system starts
- Architect security
- Develop security
- Operate securely
- Plan for failure
- Retire securely