Integrating the NICE Framework, Re-Authorization, and Security Program Results

Dr. George C. Moore
Chief Computer Scientist, Department of State

FISSEA Conference 28-Mar-2012

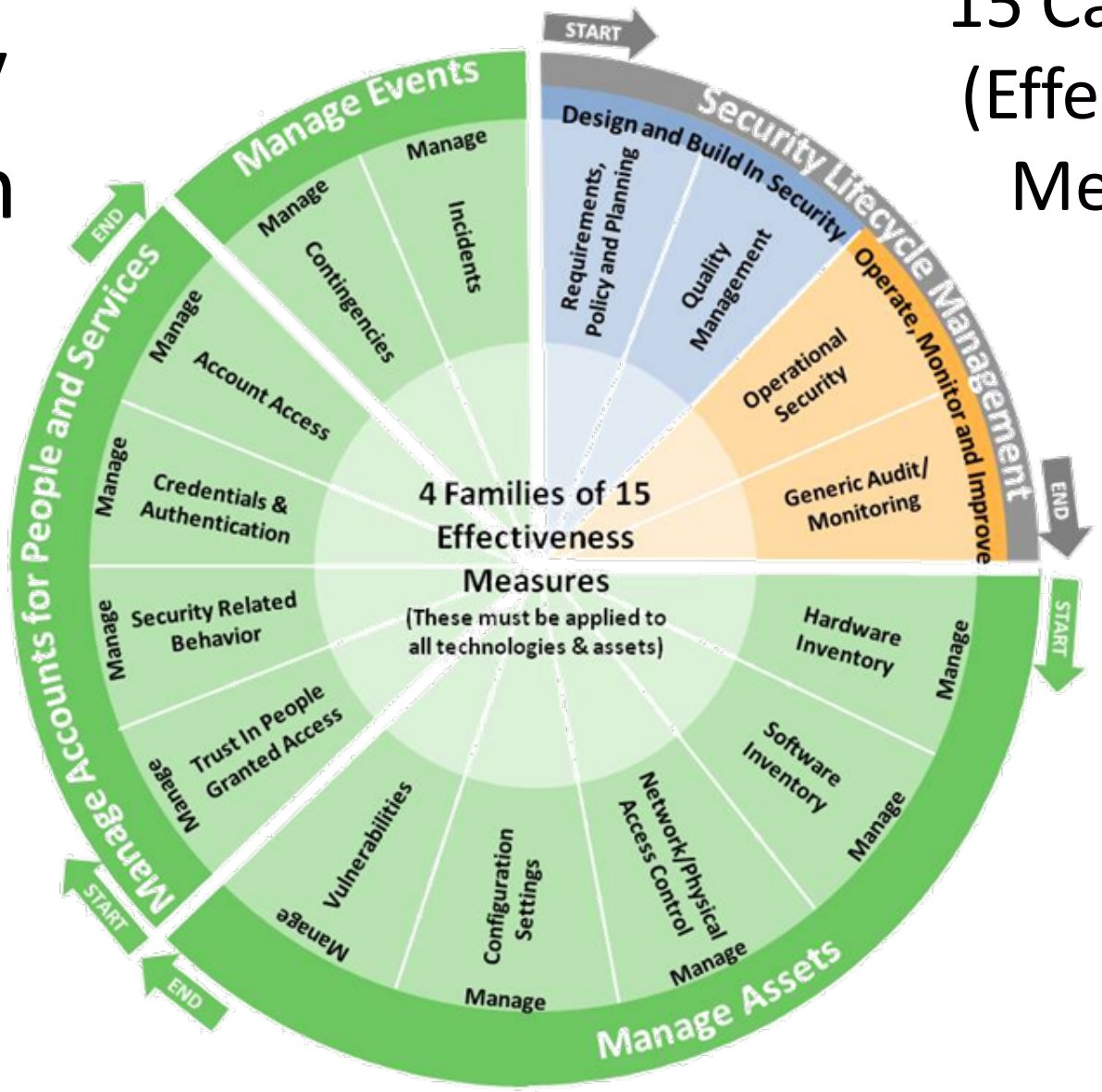
Why Security?

- Traditional security programs are based around inputs, not results.
  - Control Catalogues (no one WANTS controls).
  - Test Activity (you can't test in quality).
  - Where's the beef?
- This makes it hard to justify the program and measure its performance.
- **Good training is also based around behavioral objectives – knowing what you want the training to achieve.**

Toward Security Program RESULTS

- Work by members of the Federal Continuous Monitoring Working Group.
- Analyzed NIST SP 800-53 and SANS Critical Security Control to create “capabilities”.
- **Identify Results (“capabilities”) of 800-53 Families and SANS controls**
  - **Attack types to be blocked**
  - **Capability Statements: What must be achieved to block attacks**
- **Validation by detailed mapping of 800-53 controls to RESULTS.**

A Security Program has . .



15 Capabilities (Effectiveness Measures)

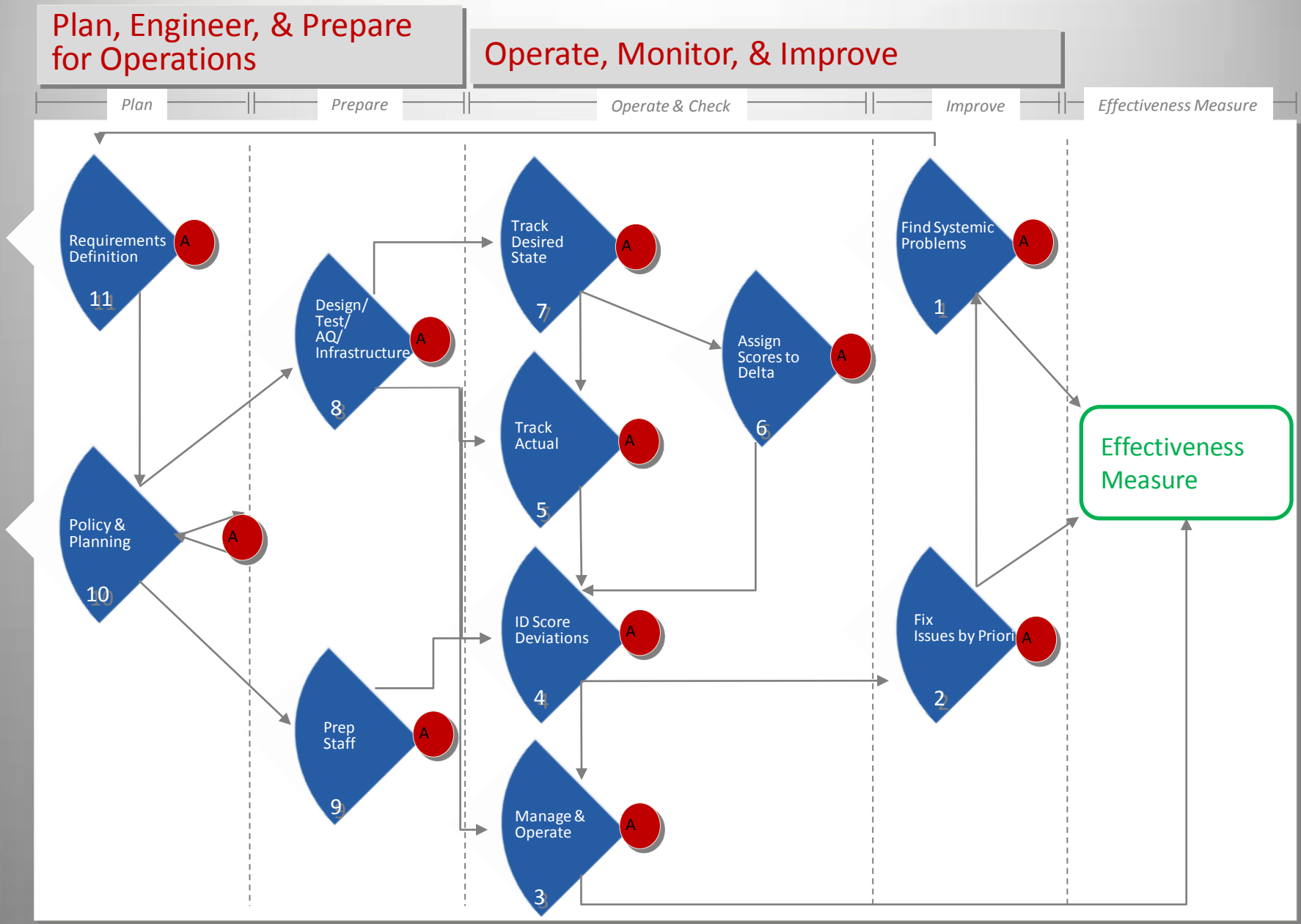
Example Effectiveness Measure for "Manage Software Inventory"

- Attacks
  - Attackers continually scan for vulnerable software and exploit it to gain control of target machines.
  - Attackers distribute hostile content on Internet-accessible (and sometimes internal) websites that exploits unpatched and improperly secured client software running on victim machines.
  - Attackers use currently infected or compromised machines to identify and exploit other vulnerable machines across an internal network.
- **Capability**
  - **Be capable of finding and removing unauthorized software with X hours to prevent these attacks.**
- **Result**
  - **Given the rate at which unauthorized software is added, the rate at which it is exploited, and the capability above, the level of resulting compromise is acceptable.**

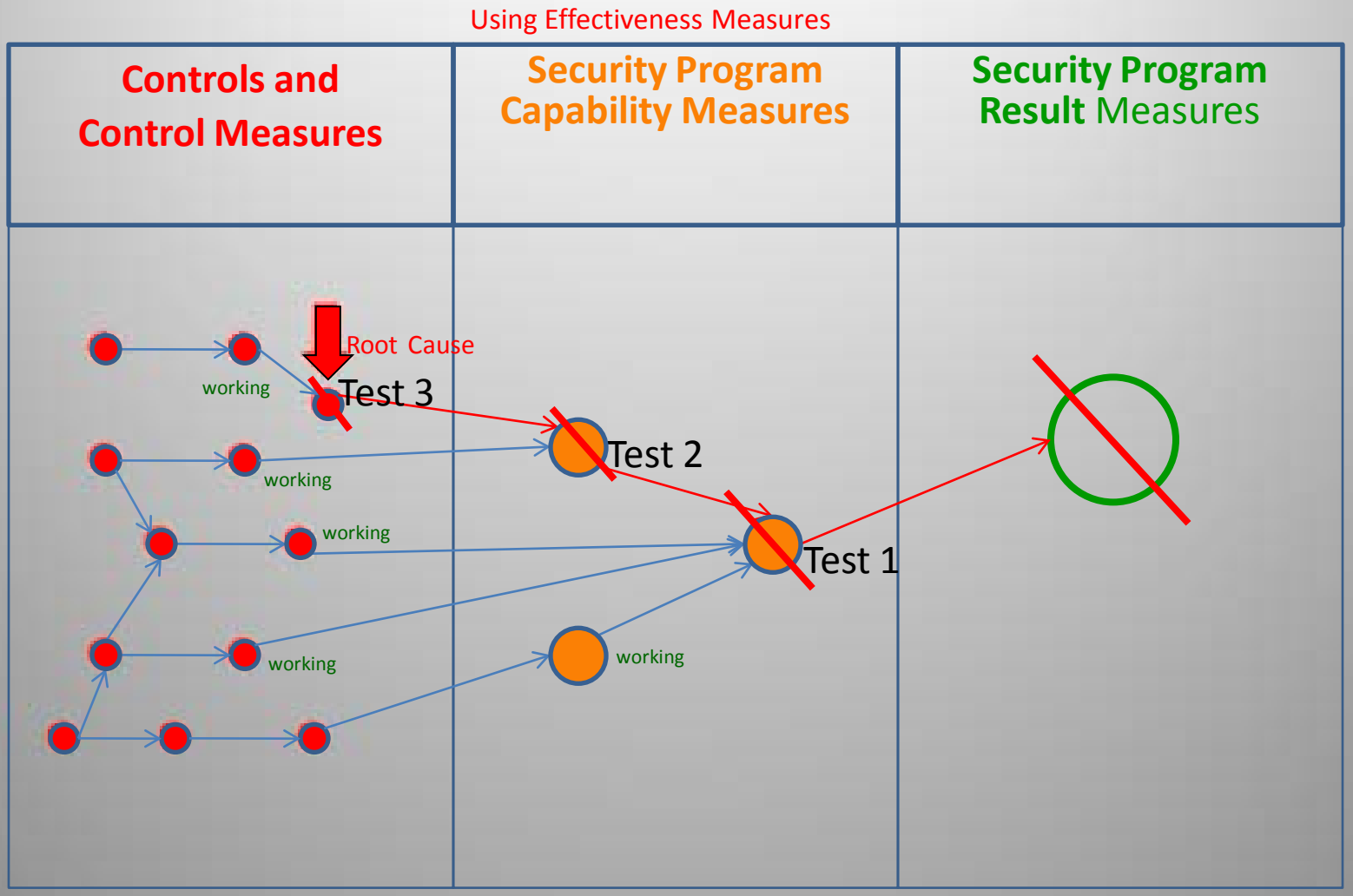
Coverage Model -- The results apply to all kinds of "assets".

	Technologies and Assets											
4 Families of 14 Effectiveness Measures **(These must be applied to all technologies and assets)**	Networks	Applications	Data	People	Wireless	Cloud	Maintenance	Media	Physical	Environmental	Malware	Etc……
Security Lifecycle Management: Design and Build in Security *Requirement, Policy and Planning (L)* *Quality Management (G1)* Operate, Monitor and Improve *Operational Security (G2)* *Generic Audit/Monitoring (F)*												
Manage Hardware and Software Assets *Manage Hardware Inventory (A)* *Manage Software Inventory (B)* *Manage Network /Physical Access Control (C)* *Manage Configuration Settings (H)* *Manage Vulnerabilities (M)*												
Manage Accounts for People and Services *Manage Trust in People Granted Access (N)* *Manage Security Related Behavior (E)* *Manage Credentials & Authentication (J)* *Manage Account Access (D)*												
Manage Events *Manage Contingencies (I)* *Manage Incidents (K)*												

Fishbone Diagrams: Value Chain



When effectiveness measures are not being met, it is not necessary to test all controls – because it is possible to trace back to the root cause



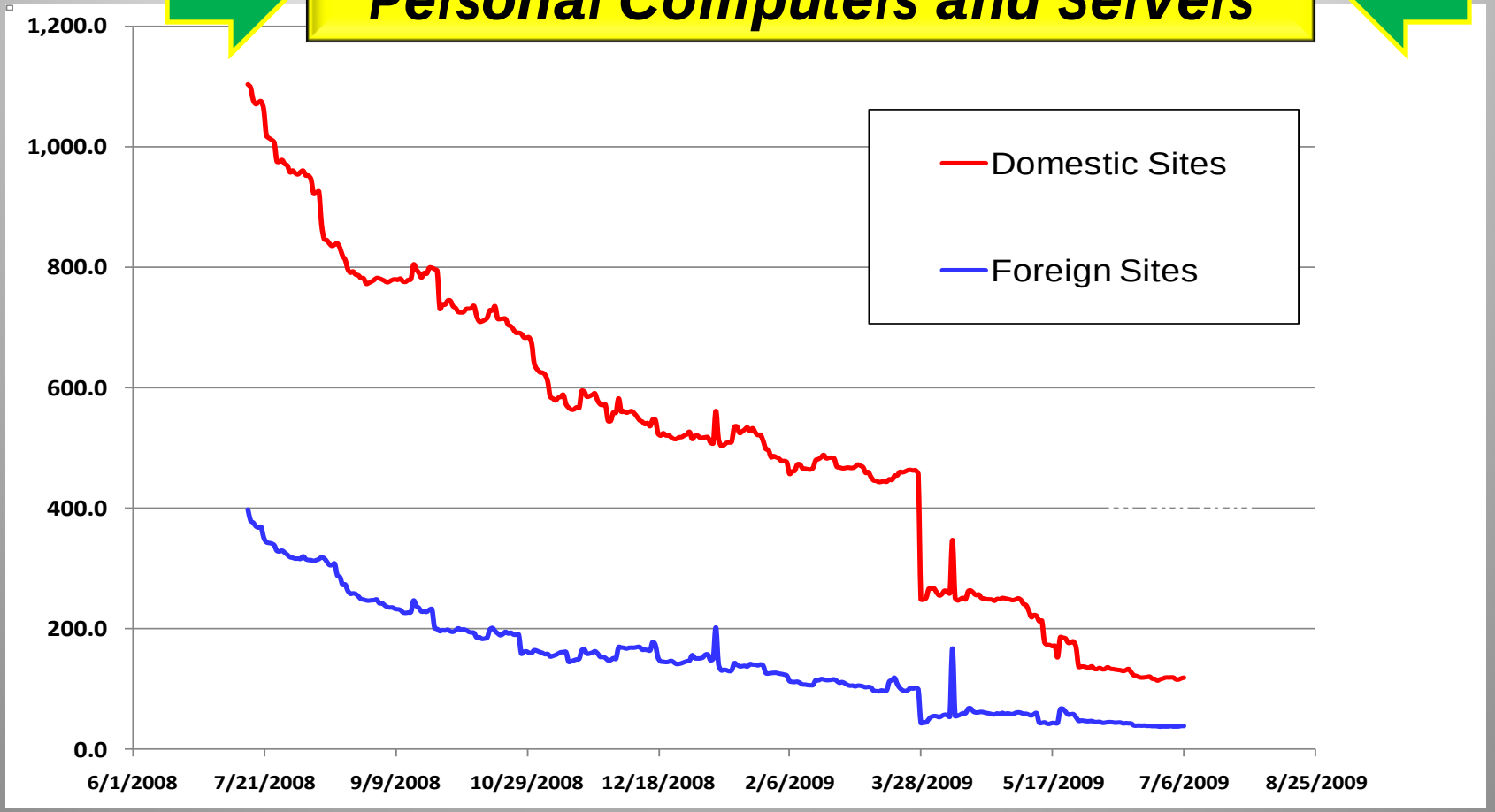
Timeliness and Completeness

- If I test all controls every three years, but the attackers find and exploit weaknesses every 2 months, how "complete" is my testing?
  - Most defects (~94.5%)* will be found and exploited before I test and find the defect.
  - So, given this assumption, the standard 3-yearly testing is only as good as "complete" testing of 5.5%* of controls every 2 months.

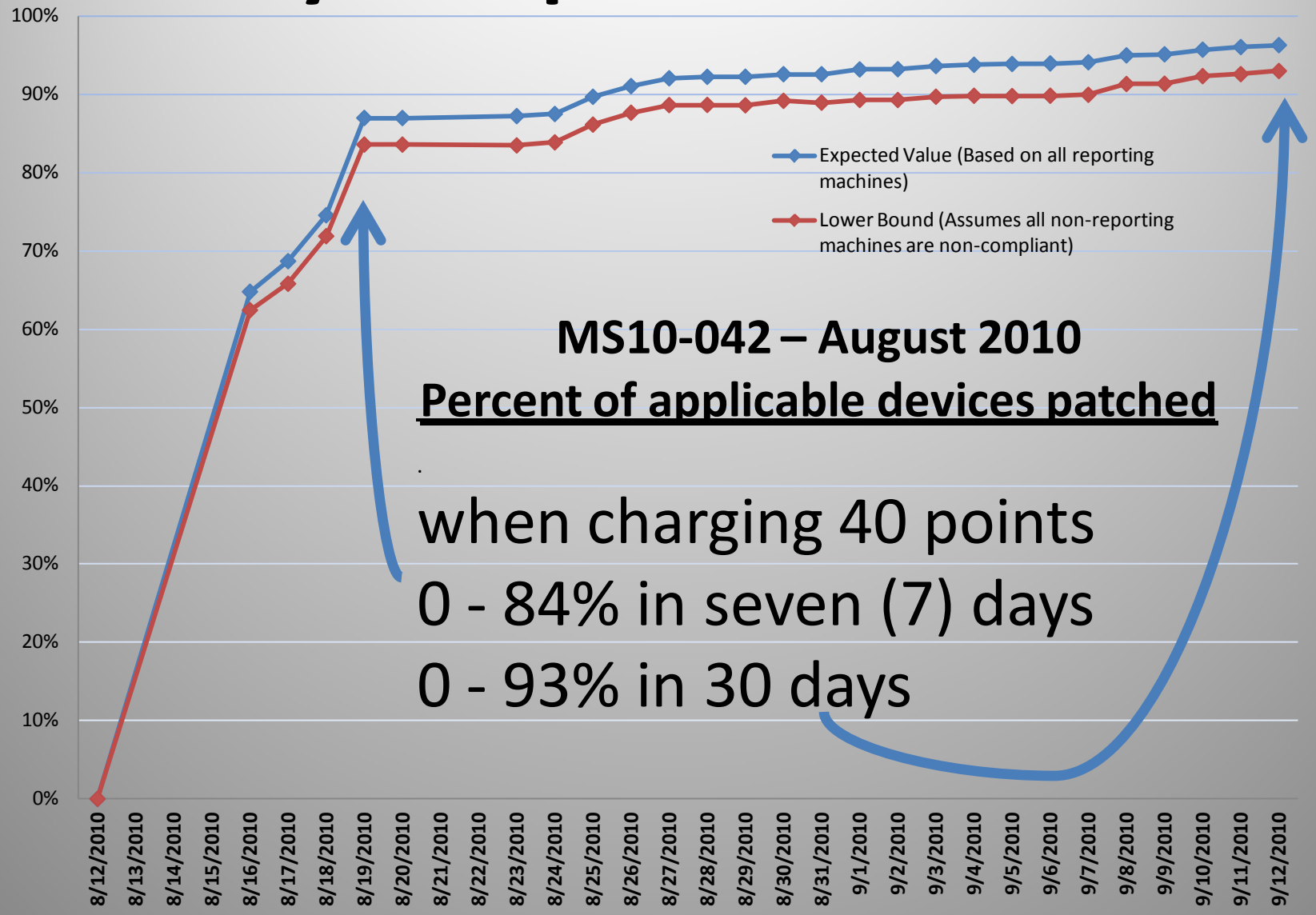
* Based on an analysis described recent paper by MIT Lincoln Labs.

Results First 12 Months

Personal Computers and Servers



Efficiency is Repeatable & Sustained



A good CMRS Program is a Training Tool

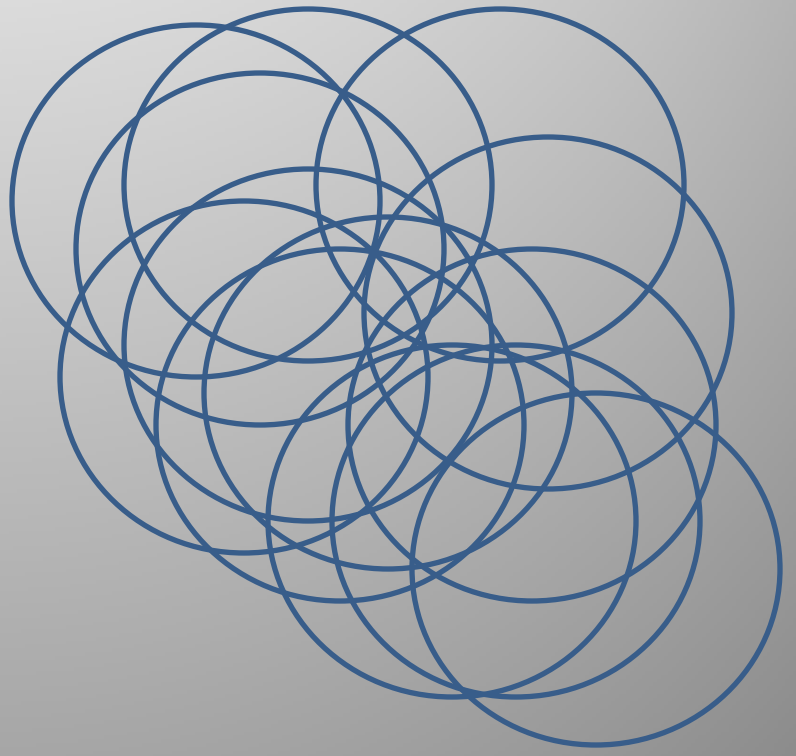
- Provides lists of this most important things to do.
- Provides detailed guidance on how to do them
- Helps non-Security Mangers set priorities
- **Continuous Monitoring with Risk Scoring becomes "just in time training" for systems administrators and executives (both are part-time security "experts").**

Problem: Security Role Definition

- A wide range of Federal groups have been laboring to define fixed cyber security roles for many years.
- Success is illusive, because organizations not only name roles differently, but they have different boundaries.
- In reality, even with a single Department, roles are defined differently in different parts.
- **Fluid, rather that static role definitions are needed** as roles flex to match organizational needed and the talents of individuals.

Significant Security Roles

- Theory – Static Roles
  - Roles are finite, clear and non-overlapping
  - They don't change much
- Practice – Fluid Roles
  - Roles are infinitely variable and overlapping
  - They move over time



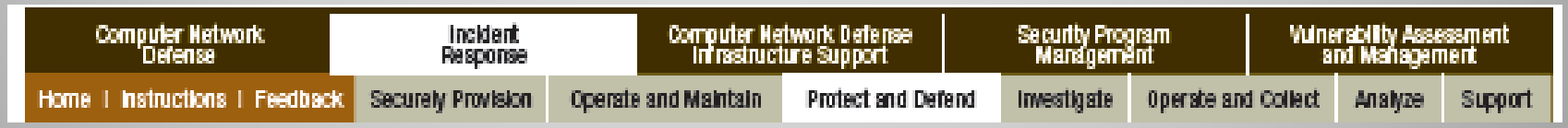
Security Effectiveness and Training

- NIST 800-53 controls are verifiably mapable to the
  - 15 Effectiveness Measures
  - 11 Fishbone Steps
  - Asset Types (some apply to all technology, others only to some technology).
- **Likewise NICE tasks and KSAs can be mapped to the same elements to help define roles.**

Example Mapping

- NICE TASK:

686	Maintain deployable Computer Network Defense toolkit (e.g., specialized Computer Network Defense software/hardware) to support incident response team mission



- **Mapping:**
  - **Capability: Incident Response**
  - **Fishbone: Provide Infrastructure**
  - **Asset: Network**
  - **Question: Is infrastructure needed for other asset types?**

Mapping provides easy role definition

- Use Case 1: A manager wants to construct a position description for a new position with special security roles. Needs tasks and KSAs for that role
- Use Case 2: A person in a specific position needs to find appropriate training for the tasks and KSAs related to their work.
- **Solution: Specify the combination of**
  - **Capabilities**
  - **Fishbone Steps**
  - **Asset Types**

  Involved in the work, THEN the mapping database will present the most relevant:
  - **Tasks**
  - **KSAs**
  - **Training Modules**

  For that work.

Conclusions

- Security is moving from:
  - Just controls **to capabilities and results**
  - Just completeness **to be more timely**
  - **Periodic Testing to Continuous monitoring and Risk Scoring to provide OTJ just-in-time training for system administrators**
  - **Fixed Roles to using mappings of capabilities, fishbones, and asset types involved in a fluid role, to identify appropriate tasks, KSAs and related training.**