Technical Foundations for Continuous Security Monitoring

9/29/2010

Peter Mell, David Waltermire, Harold Booth
NIST Senior Computer Scientists
Disclaimer and Caveats

- This presentation explores emerging and notional ideas for continuous monitoring technical foundations.
- Application to existing laws, policy, and guidance is intentionally avoided (e.g., FISMA).
- There exists NO implied policy or even NIST guidance in this presentation.
Continuous Monitoring (CM)
Presentation Contents

• Section 1: Conceptual Design Level
  – Definition, Essential Characteristics, Maturity Model, and Enterprise Architecture
• Section 2: Technical Design Level
  – Subcomponent Model, Technical Architecture
• Section 3: Implementation Design Level
  – Interfaces
  – Communication models
  – Derived test requirements (DTRs)
Providing a Layered Understanding
Driving from definitions to product testing requirements

- Definition
  - Essential Characteristics
    - Maturity Model
      - Enterprise Architecture
      - Subsystem Model
        - Technical Architecture
      - Interface Specifications
      - Communication Specifications
  - Testing Requirements
Section 1: Conceptual Design Level

- CM Definition
- Essential Characteristics
- Maturity Model
- Enterprise Architecture
Continuous Monitoring is a risk management approach to cybersecurity that maintains an accurate picture of an organization’s security risk posture, provides visibility into assets, and leverages use of automated data feeds to quantify risk, ensure effectiveness of security controls, and implement prioritized remedies.

The purpose of providing this definition is to enable us to determine the technical requirements for a CM reference architecture.
Derived CM Characteristics:

- Maintains an accurate picture of an organization’s security risk posture
- Provides visibility into assets
- Leverages automated data feeds
- Quantifies risk
- Ensures continued effectiveness of security controls
- Informs automated or human-assisted implementation of remediation
- Enables prioritization of remedies
Possible domains that CM could support

- Asset Management
- Configuration Management
- Event Management
- Incident Management
- Information Management
- License Management
- Malware Detection and Remedy
- Network Management
- Patch Management
- Software Assurance
- Vulnerability Management
Ways to Achieve CM in Your Organization

- Create ad-hoc system
  - Integrating vendor solutions to create a CM capability
  - Duplicating the work and repeating the mistakes of others
- Procure entire CM solutions from a single vendor
  - Locking into a solution that will be strong in some areas and weak in others
- Leverage a **CM technical reference architecture** and **related security standards** (e.g., SCAP)
  - Use your existing security products
  - Reduce integration costs
  - Combine best of breed solutions
Notional Maturity Model for Continuous Monitoring

from a technical maturity perspective
CM Maturity Levels 0-3

• Level 0: **Manual Assessment**
  – Security assessments lack automated solutions

• Level 1: **Automated Scanning**
  – Decentralized use of automated scanning tools
    • Either provided centrally or acquired per system
  – Reports generated independently for each system

• Level 2: **Standardized Measurement**
  – Reports generated independently for each system
  – Enable use of standardized content (e.g., USGCB/FDCC, CVE, CCE)

• Level 3: **Continuous Monitoring**
  – Reports generated independently for each system
  – Federated control of automated scanning tools
  – Diverse security measurements aggregated into risk scores
    • Requires standard measurement system, metrics, and enumerations
  – Comparative risk scoring is provided to enterprise (e.g., through dashboards)
  – Remediation is motivated and tracked by distribution of risk scores
CM Maturity Levels 4-5

- **Maturity level 4:** Adaptable Continuous Monitoring
  - Enable plug-and-play CM components (e.g., using standard interfaces)
  - Result formats are standardized
  - Centrally initiated ad-hoc automated querying throughout enterprise on diverse devices (e.g., for the latest US-CERT alert)

- **Maturity level 5:** Continuous Management
  - Risk remedy capabilities added (both mitigation and remediation)
  - Centrally initiated ad-hoc automated remediation throughout enterprise on diverse devices (with review and approval of individual operating units)
    - Requires adoption of standards based remediation languages, policy devices, and validated tools
# Maturity Model Level Characteristics

<table>
<thead>
<tr>
<th></th>
<th>Level 0</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interfaces</strong></td>
<td>Undefined</td>
<td>Unused</td>
<td>Unused</td>
<td>Proprietary</td>
<td>Standardized</td>
<td>Standardized</td>
</tr>
<tr>
<td><strong>Security Check</strong></td>
<td>Prose</td>
<td>Proprietary</td>
<td>Some Standardization</td>
<td>Some Standardization</td>
<td>Fully Standardized</td>
<td>Fully Standardized</td>
</tr>
<tr>
<td><strong>Content Format</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reporting</strong></td>
<td>Ad hoc</td>
<td>Proprietary and not Integrated</td>
<td>Proprietary and not Integrated</td>
<td>Coarse integration / some standardization</td>
<td>Standardized integration</td>
<td>Standardized integration</td>
</tr>
<tr>
<td><strong>Remedies</strong></td>
<td>Manual</td>
<td>Manual or Proprietary</td>
<td>Manual or Proprietary</td>
<td>Manual or Proprietary</td>
<td>Manual or Proprietary</td>
<td>Standardized Automation</td>
</tr>
</tbody>
</table>
Important CM solution goals:

• Component based approach
  – Based on a standardized reference architecture
  – Solutions from multiple vendors can be combined together to create a CM solution
• Standard-based for interoperability and scoring consistency
  – Languages
    • Using the same machine-readable expressions for checking and remediating machine state (e.g., FDCC policy)
  – Metrics
    • Using the same equations for risk calculations
  – Nomenclatures
    • Using the same names for vulnerabilities, assets, configuration issues, and remediation options.
• Mathematically rigorous scoring approach
  – Motivational scoring is important
  – True risk calculations are also needed
Notional CM Enterprise Architecture

- This shows an enterprise architecture view, not a technology focus view

Diagram derived from other government work
Section 2: Technical Architecture Design Level

- Technical Models
- Subcomponent Design
- Interface Identification
DHS Continuous Asset Evaluation, Situational Awareness, and Risk Scoring (CAESARS) Reference Architecture
Notional Ideas for Enhanced Models

- Multiple subsystems instances (already in CAESARS)
- CM multi-instance capability (e.g., hierarchical tiers)
- Interface definitions
- Enhanced communication payload definitions
- Specifications describing subcomponent functionality
  - Could lead to a product validation program or agency procurement (e.g., DHS ISSLOB)
Hierarchical Federated Architecture

- Large organizations will have more than one CM instance
- CM instances are usually arranged in a logical hierarchy
  - Aggregated reports travel up the tree
  - Data calls and configuration requirements travel down the tree
- Often CM instances have a degree of autonomy resulting in a federated style of communication
  - Each instance may have approval authority on directives from higher levels
- Lateral communication in the tree is also possible
Organizations may have multiple CM instances

CM System Instance Subsystems
- 1+ Presentation / Reporting Subsystem
- 1+ Analysis / Risk Scoring Subsystem
- 1 Data Aggregation Subsystem
- 1+ Sensor Subsystem
- 0-1 Content Subsystem (need 1 somewhere in enterprise)
- 0-1 Task Manager Subsystem (optional but valuable)

Outside entities
- National Vulnerability database (NVD)
- U.S. Government Configuration Baseline (USGCB)
Why Have a Task Manager Subsystem?

• Single CM Instance
  – Orchestrates scanning, aggregation and reporting activities within the system
    • Harness input from diverse security devices
  – Enable ad hoc queries from dashboard
    • Automatically retrieve data not already in data aggregation subsystem

• Multi-instance Federated Hierarchical Architecture
  – Avoid tendency (and possibly need) to aggregate all data up through all tiers
  – Enable higher tiers to request specific data from lower tiers
  – Provide policy management of requests entering a tier
  – Could enable a “big easy” button with safety controls and tiered human review and approval
Why Have a Content Subsystem?

- Enables both organization-wide and locally-scoped content
- Holds machine readable security baselines (e.g., Federal Desktop Core Configuration)
- Allows organizations to tailor or augment baselines for their own needs
- Typical baseline standards include:
  - Extensible Configuration Checklist Description Format (XCCDF)
  - Open Vulnerability and Assessment Language (OVAL)
  - Open Checklist Interactive Language (OCIL)
- Content subsystem implementation approaches
  - 1 content subsystem for entire organization
  - 1 content subsystem per CM instance
    - Adds complexity to score aggregation (apples vs. oranges)
  - Hybrid model (allow only certain tiers to customize)
Notional Multi-instance CM Architecture

- This view shows the relationship between CM instances
- These interfaces enable the hierarchical federated CM architecture
Notional CM Instance Architecture
Section 3: Implementation Design Level

- Interface Specifications
- Communication Models
- Derived Test Requirements
Challenges in Defining Interfaces (payload + communication mechanism)

- I4, I8, and I10: Focus of this work (I8 substantially addressed through DHS CAESARS)
- I6: National Vulnerability Database (NVD) XML file and WSDL interfaces are defined
- I1, I3, and I5: No current standards exist for arbitrary data retrieval
  - Use of SQL would require mandating a particular database schema to be implemented within products
  - Interfaces could be left proprietary in the short term and we could watch for best of breed solutions to appear from vendors
  - Refinement of Policy Language for Assessments Results Reporting (PLARR) to address part of the problem
- I7, I9, I11: Future work on multi-tier request and security automation content propagation
Notional Multi-instance CM Architecture

- This view shows the relationship between CM instances
- These interfaces enable the hierarchical federated CM architecture

Interface and Payload Specifications:
- Existing
- Current Focus
- Proprietary/Future Focus
Notional Interface Overview: I10

- **Interfaces:**
  - Service Oriented Architecture
  - Web Services Description Language (WSDL) direct connection
  - Enterprise Service Bus
  - Other interfaces??

- **XML communication envelope: Asset Reporting Format (ARF)**

- **XML payload options:**
  - USG XML schema data (based on USG agreed upon metrics)
  - SCAP XML (e.g., XCCDF results, OVAL results)
  - Vendor proprietary XML

- **Use of proprietary payloads may require additional integration and loss of plug and play compatibility**
Notional CM Instance Architecture

Interface and Payload Specifications:
- Existing
- Current Focus
- Proprietary/Future Focus

CM System Instance Model

Data Access Capability
- Task Manager (optional)
  - CM Task Manager
- Data Aggregation
  - CM Data Aggregator

Content (optional)
- CM Content Repository

Situational Awareness Capability
- Analysis/Risk Scoring
  - Risk Analysis Engine
  - Risk Scoring Engine
- Presentation/Reporting
  - Presentation/Dashboard Engine
  - Ad-hoc Capability

Sensors
- Sensor Manager
- Authoritative Data Repository

*: An instance of the content repository is required in some part of the overall CM architecture

Other Content Providers
- National Vulnerability Database
- U.S. Government Configuration Baseline
Notional Interface Overview: I8

- **Interfaces:**
  - Service Oriented Architecture
    - WSDL direct connection
    - Enterprise Service Bus
  - Other interfaces??
- **XML communication envelope: ARF**
- **XML payload options:**
  - Need to define standards-based payload(s) to support all sensor types
    - System configuration management
    - Anti-virus
    - Web vulnerability scanner
    - Database vulnerability scanner
    - Unauthenticated vulnerability scanner
    - Authenticated vulnerability and patch scanner
    - Authenticated configuration scanner
    - Network configuration management tools
    - Federal Desktop Core Configuration scanner
  - Leverage Security Content Automation Protocol XML (e.g., XCCDF results, OVAL results)
  - Allow vendor proprietary XML??
Notional and Under Development Communication Models
Closing Thoughts

- There exists great momentum surrounding CM (both executive level and grass roots)
  - Dashboards, “big easy” buttons, aggregated reporting of technical metrics
- Agencies can leverage their existing security tools to evolve towards an automated CM solution
  - Enhance their own capability and meet upcoming reporting demands
- Reference architectures
  - Can reduce integration efforts
  - Enable CM plug-and-play component capabilities
    - Product validation and procurement programs can assist with tool adoption of necessary technical specifications
    - Focus agencies on evolving toward the full potential of CM
- The long term vision will take time and effort, but significant gains are achievable today.
Acknowledgements and Credit

- Much of this was inspired and encouraged by others
  - Information Security and Identity Management Committee (ISIMC) CM working group
  - DHS Federal Network Security (Cyberscope and CAESARS)
  - NSA Information Assurance Directorate (IAD)
  - NIST Security Content Automation Protocol (SCAP) team
  - MITRE McLean CAESARS team
  - MITRE Bedford SCAP team
Summary and Questions

Presenters:

Peter Mell
NIST Senior Computer Scientist
301-975-5572
peter.mell@nist.gov

David Waltermire
NIST Senior Computer Scientist
301-975-3390
david.waltermire@nist.gov