Continuous Diagnostics and Mitigation

Readiness and Planning Guide for Asset-Based CDM Security Capabilities

Version 1.0

January 29, 2016

Homeland Security

Federal Network Resilience

Revision/Change Record

Revision	Date	Revision/Change Description	Pages Affected
Version 1.0	January 29, 2016	Final effective 1/29/16	All

Table of Contents

Readiness and Planning Guide for Asset-Based CDM Security Capabilities

1 Purpose

This document proposes practical considerations for departments and agencies (D/As) to leverage when implementing the Continuous Diagnostics and Mitigation (CDM) Program asset-based security capabilities. Asset-based security capabilities consist of

- Hardware Asset Management (HWAM)
- Software Asset Management (SWAM)
- Configuration Settings Management (CSM)
- Vulnerability Management (VUL)

D/As, in coordination with the Continuous Monitoring as a Service (CMaaS) Provider, may use this document to assist with integrating CDM Program processes and activities into an established information security continuous monitoring (ISCM) program. This guide identifies practices supportive to implementation of CDM asset-based security capabilities.

The goal of this document is to provide D/As with a set of readiness considerations to support discussion with the CMaaS Provider either before or during implementation of the CDM asset-based security capabilities. It is not intended to be a CDM design or implementation framework, which is the responsibility of the selected CMaaS provider.

Additionally, this document is not a Department of Homeland Security (DHS) data call. Answers and/or notations made to the readiness considerations are not for DHS use or collection; rather they serve to improve information exchange between D/A personnel and CMaaS providers as preparation for CDM implementation.

Lastly, this guide does not mandate CDM readiness requirements and does not represent an exhaustive list of every concern a D/A may encounter when preparing to implement CDM asset-based security capabilities. It is designed to assist D/As by identifying the essential practices that are foundational for a more successful implementation of CDM asset-based security capabilities. It is anticipated that D/As could utilize this information in alignment with their current ISCM program, Federal Information Security Modernization Act (FISMA) compliance program, or other cybersecurity management structure.

Note that DHS recognizes that not all roles, processes, or procedures are directly managed by the D/A, especially for small and micro-sized D/As. Therefore, D/As are encouraged to customize this guide as appropriate to improve the effectiveness and efficiency of their respective CDM implementation.

2 Background

The CDM Program is a dynamic approach to fortifying the cybersecurity of government networks and systems. CDM provides federal departments and agencies with security capabilities and tools that identify cybersecurity risks on an ongoing basis, prioritize these risks based on potential impacts, and enable cybersecurity personnel to mitigate the most significant problems first. Congress established the CDM Program to provide adequate, risk-based, and cost-effective cybersecurity and more efficiently allocate cybersecurity resources.

The CDM Program enables government entities to expand their continuous diagnostic security capabilities by increasing their network sensor capacity, automating sensor collections, and prioritizing risk alerts.

CDM offers commercial off-the-shelf (COTS) tools, with robust terms for technical modernization as threats change. Agency-installed sensors perform an automated search for known cyber flaws. Results feed into a local dashboard that produces customized reports, alerting network managers to their most critical cyber risks based on standardized and weighted risk scores. Prioritized alerts enable agencies to efficiently allocate resources based on the severity of the risk. Progress reports track results, which can be used to compare security posture among department/agency networks. Summary information can feed into an enterprise-level dashboard to inform and situational awareness into cybersecurity risk posture across the federal government.

2.1 Relationship to other CDM Guides

The *Roles and Responsibilities Guide* proposes roles and responsibilities for organizations implementing the CDM Program.

The *CDM Readiness and Planning Guides* propose questions for D/As to consider before implementing CDM Program security capabilities.

The *Implementation Guides* describe the recommended operational practices for CDM security capabilities.

3 Intended Audience

This guide is intended to assist individuals responsible for the design, development, implementation, operation, maintenance, and disposal of federal information systems and specifically the implementation and operation of the CDM Program implementation to include individuals specifically responsible for the following:

Information system development and integration responsibilities to include but not limited to program managers, information system developers, information systems integrators, enterprise architects, network architects, and information security architects

Information system and/or security management/oversight responsibilities to include but not limited to senior leaders, risk executives, authorizing officials, chief information officers, and chief information security officers

Information system and security control assessment and monitoring responsibilities to include but not limited to system evaluators, assessment teams, independent verification and validation assessors, auditors, or information system owners

Information security implementation and operational responsibilities to include but not limited to information system owners and administrators, common control providers, information owners/stewards, mission/business owners, information security architects, and information system security engineers/officers

4 CDM Asset-Based Security Capabilities Scope

The CDM Program covers 15 continuous diagnostic security capabilities. The first phase of CDM focuses on endpoint integrity: management of hardware and software assets, configuration management, and vulnerability management, which are foundational security capabilities to protect systems and data. Phases 2 and 3 are being further defined to include Least Privilege and Infrastructure Integrity, and Boundary Protection and Event Management, respectively.

5 CDM Asset-Based Security Capability Practices

The Readiness and Planning Guide for Implementing CDM Asset-Based Security Capabilities defines a practice as a cybersecurity-related activity or process supportive to one or more CDM asset-based security capabilities. By identifying practices, this guide helps a D/A leverage its current business processes to implement CDM asset-based security capabilities.

This CDM guide references the *NIST Framework for Improving Critical Infrastructure Cybersecurity* (hereafter referred to as the *Cybersecurity Framework*) because of its wide adoption.

The *Cybersecurity Framework* was developed in response to Executive Order 13636, "Improving Critical Infrastructure Cybersecurity," issued by President Obama on February 12, 2013. It is a framework that is technology neutral and extensible to allow for flexibility and innovation, and it relies on a variety of existing standards, guidelines, and practices. It also uses risk-management processes to enable organizations to inform and prioritize decisions allowing for risk-based implementation.

The *Cybersecurity Framework* is composed of three parts: the Framework Core, the Framework Implementation Tiers, and the Framework Profiles. This workbook focuses on the Framework Core, specifically the subcategories within the Framework Core. Subcategories are used within the *Cybersecurity Framework* to further divide a category into specific outcomes of technical and/or management activities. They provide a set of results that, while not exhaustive, help support achievement of the outcomes in each category.

The Readiness and Planning Guide for Implementing CDM Asset-Based Security Capabilities identifies a *Cybersecurity Framework* subcategory to represent a foundational and security capability-specific practice. This subcategory was defined because it was understood that organizations are currently performing activities directly supportive to a *Cybersecurity Framework* subcategory and each of the selected *Cybersecurity Framework* subcategories is directly supportive to implementing asset-based security capabilities of the CDM Program. Only *Cybersecurity Framework* subcategories directly supportive to a foundational or CDM asset-based security capability desired state specification were identified as practices.

The practices have been divided into two groups: foundational and capability-specific practices. Foundational practices are those practices common to two or more CDM asset-based security capabilities. Capability-specific practices are those practices unique to a specific CDM asset-based security capability. This *CDM Guide* is organized by a practice followed by a series of considerations; each consideration illuminates an aspect of the practice supportive to CDM asset-based security capabilities in general or to a specific security capability desired state specification.

Considerations are intended to help an organization self-identify business processes and management activities necessary for CDM asset-based implementation.

It is understood that D/As are dissimilar with respect to size and security capabilities and that they will inherently be at different stages of readiness to implement CDM asset-based security capabilities. This guide has focused on assisting the D/As with preparing for implementation by highlighting areas that should be considered before implementing CDM asset-based security capabilities. The D/A will be better prepared to implement CDM asset-based security capabilities based on how completely they are able to answer the practice consideration.

Note that the ability to fully answer each practice consideration is not required to actually start the implementation process of CDM phase 1 security capabilities. The D/A may use this guide in the manner which best supports its preparation activities.

Suggestions for answering considerations include

- Providing full-text answers for each consideration
- Answering considerations by referencing other D/A documents such as policies or standard operating procedures
- Providing hyperlinks to documents in repositories such as SharePoint portals
- Answering in a separate document, presentation, database, or SharePoint portal, such as a list or wiki

Additionally, the following are the top four recommended actions or takeaways a D/A can implement to improve readiness for successful implementation of CDM asset-based security capabilities:

- Enable and empower different components within the organization to communicate about CDM readiness and issues, to include establishing CDM intra-agency working groups across components.
- Identify key stakeholders and points of contact (POCs) for the CDM project (e.g., engineering and integration, security, network, information assurance, system administrators, and decision makers) to be prepared for the D/A CDM kickoff meeting.
- Enable organization plans to manage the CDM implementation scope.
- Identify roles and processes that support D/A responsibilities for CDM policy, process, and desired state management.

5.1 Foundational Practice

The following practice provides a foundation for all CDM asset-based security capabilities.

5.1.1 Foundational Practice 1: Information security roles and responsibilities are coordinated and aligned with internal roles and external partners.

This practice focuses the D/A to consider existing roles and responsibilities inherently supportive to CDM asset-based security capability implementation. Examples include roles and responsibilities defined by an organization's ISCM program or other similar governance-related programs.

Table 1: Considerations Regarding Intra-Organization Coordination for CDM

Considerations	Notations and Comments
How do our components/offices communicate about CDM readiness, concerns, and issues?	
Do we (or should we) have intra-agency working groups supporting CDM?	
What is our coverage and scope of CDM working groups?	
How often do our intra-agency CDM working groups meet?	
Are our CDM intra-agency working groups discussing topics such as tool employment issues, best practices, risks, challenges, change control, etc.?	

Table 2: Considerations Regarding Roles and Staff Availability

Considerations	Notations and Comments
Who are our key stakeholders for the CDM kickoff meeting?	
What is the availability of our staff responsible for explaining, verifying, and validating the scope of the network for CDM and what is on those networks?	
How up to date and ready is our network architecture documentation for CDM?	
Do we have a CDM training and awareness program for our internal staff and key stakeholders performed?	
Have we designated personnel to support the CDM Program?	
Who are our CDM Program change control board (CCB) POCs?	
How does our CCB process work?	
Who are our POCs for the CDM project (security, network, IA, system administrators, and decision makers)?	
Who are our key stakeholders for the discovery meetings, including as-is discovery?	

Table 3: Considerations Regarding the Processes Supporting the CDM Program in General

Considerations	Notations and Comments
What is the process for acquiring logical and physical access to our assets for the CMaaS provider? Examples include vetting/suitability process, and onboarding of required staff.	
How is the authorization and accreditation process currently implemented within our organization?	
Who are the authorization and accreditation process points of contact within our organization?	
How are the security requirements managed for the authorization and accreditation process?	

5.2 Security Capability-Specific Practices

The following subsections address specific practices for each of the asset-based CDM security capabilities.

5.2.1 Manage Hardware Assets (HWAM) Security Capability

The HWAM security capability addresses attacks that seek to exploit unauthorized and unmanaged hardware assets. The HWAM security capability gives organizations visibility into the hardware operating on their network(s) and ensures that they are identified, authorized, and managed to reduce vulnerabilities and thwart attacks. It also provides a view of hardware asset management responsibility such that prioritized defects can be presented to the responsible party for mitigation actions and risk acceptance decisions.

The operational execution of the HWAM security capability is dependent on the following desired state specifications:

- Only devices (i.e., hardware assets) in the authorized hardware inventory are on the network.
- All authorized hardware assets are in the authorized hardware inventory.
- All authorized hardware assets are assigned to a manager.

Based on the HWAM desired state specifications, the following practices support a D/A in successfully implementing the HWAM security capability:

- HWAM Practice 1: All authorized devices (i.e., hardware assets) are in the hardware asset inventory.
- HWAM Practice 2: Only authorized hardware assets are allowed to be on the network.
- HWAM Practice 3: All authorized hardware assets have a manager assigned to them.

5.2.1.1 HWAM Practice 1: All authorized devices (i.e., hardware assets) are in the hardware asset inventory.

How an organization collects and stores hardware asset inventory is vital to the operations of the HWAM security capability. Therefore, HWAM Practice 1 addresses the organization's use of asset management databases and the processes used to update, maintain, and ensure the accuracy of data in its hardware inventories.

The following considerations can be used to assess an organization's ability to fully implement HWAM Practice 1. These considerations are designed to provide insight on existing organizational policies and processes related to the HWAM security capability.

Table 4: Considerations Related to the Hardware Asset Inventory

Considerations	Notations and Comments
Does our organization have a centralized or distributed hardware inventory?	
Where do we store the hardware inventory?	
Do we have established policies, practices, and procedures to inventory attributes on hardware assets?	
How readily available is our hardware inventory for electronic transfer and automation?	

Table 5: Considerations Regarding the Hardware Inventory Data

Considerations	Notations and Comments
Does our organization have a policy specifying what attributes must be inventoried for each hardware asset? Attribute examples include IP address, MAC address, manager, location, etc.	
What guidance or process do we use to determine how hardware asset attributes are captured and documented? In other words, if we do not have a policy, what determines the attribute collection method?	
How do we document our hardware asset inventory?	
How do we uniquely identify a hardware asset?	

Table 6: Considerations Related to the Hardware Authorization Process

Considerations	Notations and Comments
What roles do we have that are involved in authorizing a hardware asset prior to it joining the network?	
How long does it take for us to update our hardware assets inventory once a new hardware asset is authorized?	
Do we have a process for authorizing a hardware asset prior to it joining the network?	

Considerations	Notations and Comments
What process, procedures, and tools do we use or evaluate in our hardware asset authorization process?	
When and how often (on average) does our hardware asset authorization process take place?	
How are the outputs of our hardware asset authorization process documented?	

Table 7: Considerations regarding the operational use of the hardware asset inventory

Considerations	Notations and Comments
How consistent are our hardware asset management processes across the different parts of our organization?	
What roles are involved in validating our hardware asset inventory?	
How long does it typically take to remove a hardware asset from our network once authorization is expired or is revoked?	
What processes and procedures do we follow when an unauthorized hardware asset is discovered on the network?	
What roles are involved in hardware asset operations and management?	

5.2.1.2 HWAM Practice 2: Only authorized hardware assets are allowed to be on the network.

HWAM Practice 2 focuses on hardware assets for which the organization is responsible to secure, but does not directly host. Examples include cloud services or employee-provided hardware assets such as mobile devices (to include D/A-managed containers on personal mobile devices).

Table 8: Considerations Related to Externally Managed Hardware Assets

Considerations	Notations and Comments
What process do we use for listing externally managed hardware assets? For example, BYOD or third-party assets.	
Do we allow externally managed hardware assets to connect to the network?	
What process do we follow before allowing externally managed hardware assets to access our network services?	

5.2.1.3 HWAM Practice 3: All authorized hardware assets have a manager assigned to them.

HWAM Practice 3 is built on ensuring that hardware assets are managed and maintained through a formal process and assignment of responsibilities throughout their lifecycles.

Table 9: Considerations Related to Formal Management of Hardware Assets

Considerations	Notations and Comments
Which components/groups across our organization own hardware assets?	
How do we identify and assign responsibility to manage hardware assets?	
Which components/groups in our organization are responsible for the administration of hardware assets?	
What process do we follow when it is determined that a hardware asset should be removed from the network?	

5.2.2 Manage Software Assets (SWAM) Security Capability

The SWAM security capability gives an organization visibility into the software that is installed and is operating on its network(s) so it can appropriately manage authorized software and remove unauthorized software. It also provides a view of software management responsibility (i.e., who patches the software, who configures the software, who decides what versions are allowed for the organization) such that prioritized defects can be presented to the responsible party for mitigation actions and risk acceptance decisions. The SWAM security capability is dependent on the existence of a set of hardware asset roles defined for the D/A and an authorized hardware inventory as developed for the HWAM security capability.

The operational execution of the SWAM security capability is dependent on the following desired state specifications:

- Only authorized software products (i.e., software assets) and executable files are installed on devices (i.e., hardware assets).
- All hardware assets are assigned or authorized for a set of hardware asset attributes, and any authorizations are revalidated on a periodic basis.
- All software assets installation and execution restriction mechanisms are deployed and configured correctly.
- All blacklists are up to date.
- All software assets not explicitly identified in a whitelist or blacklist are included in the graylist.
- All graylist software assets are investigated and deemed authorized or unauthorized within a specific period of time, and are added to the appropriate whitelist or blacklist.

Based on the SWAM desired state specifications, the following practices support a D/A in successfully implementing the SWAM security capability:

- SWAM Practice 1: Software platforms and applications within the organization are inventoried.
- SWAM Practice 2: Malicious code is detected.

- SWAM Practice 3: Integrity checking mechanisms are used to verify software, firmware, and information integrity.

5.2.2.1 SWAM Practice 1: Software platforms and applications within the organization are inventoried.

SWAM Practice 1 focuses on how the organization collects and stores information on the software assets in its software inventories and software management databases. Additionally, it pertains to how an organization updates and maintains accurate and current data about software assets in the inventory.

Table 10: Considerations Related to the Software Asset Inventory

Considerations	Notations and Comments
How do we list our software assets?	
How do we document our software assets?	
How do we store our software assets inventory list?	
Where do we store our software assets inventory list?	
What format is our software assets inventory in?	
How readily available for electronic transport/ingestion is our software assets inventory?	
Do we have an authoritative blacklist, graylist, and whitelist?	
How do we define a software asset?	
Are custom/integrated software assets properly identified as software assets?	

Table 11: Considerations Related to the Software Asset Inventory Data

Considerations	Notations and Comments
How do we validate the accuracy of our software assets inventory?	
How do we whitelist software assets?	
How do we blacklist software assets?	
How do we differentiate unique software assets?	
What attributes are collected for each software asset?	
Do our hardware assets have an authorized list of attributes and does it include software and hardware attributes?	
How are hardware asset attributes updates performed?	

Table 12: Considerations Regarding the Process of Updating the Inventory as New Software Assets are Authorized

Considerations	Notations and Comments
Is there a software assets graylist? If so, what is the policy, process, and procedure for software to be moved to the blacklist or whitelist?	
How long does it take to add software assets to the whitelist?	
How long does it take to add software assets to the blacklist?	
How do we maintain the list of software assets for each hardware asset?	
What policies and processes do we have associated with integrating hardware asset and software asset inventories?	
Do we have a process for authorizing software assets?	
How do we authorize (approve) software assets?	
How often does the software assets authorization process take place?	
How do we document the output of software asset authorization?	
Is the software asset authorization process integrated with hardware asset management?	
How long does it take to update the software asset inventory once a software asset is authorized?	
Are new software asset versions managed as a separate software asset?	

Table 13: Considerations Regarding the Operational Use of the Software Asset Inventory

Considerations	Notations and Comments
How are policy conflicts (e.g., software asset authorization levels and configuration) resolved?	
What roles are involved in authorizing software assets?	
What roles are involved in validating the software assets inventory?	
What roles are involved in software asset management?	
Is there a role for validating software asset authorization status?	
What roles are involved in license management, maintenance, procurement, change management, and logistics?	
How consistent are software asset management processes across the different parts of our organization?	
Is the software asset authorization process centralized or distributed across different parts of our organization?	
Is the complete software asset list readily available?	

5.2.2.2 SWAM Practice 2: Malicious code is detected.

Similar to SWAM Practice 1, SWAM Practice 2 focuses on software assets that the organization must keep as unauthorized and blacklisted, such as software the organization has deemed unwanted or vulnerable.

Table 14: Considerations Regarding Malicious Software Code Detection

Considerations	Notations and Comments
What is the process to determine what is or is not malicious code?	
How long does the process to determine what is or is not malicious code take?	
How do we maintain our list of malicious code?	
How do we maintain our non-malicious, organizationally defined blacklist?	
How often are known-bad blacklists updated?	
What attributes determine if a software asset belongs on the blacklist?	
Have we defined a responsible party for acting on malicious code detection?	

5.2.2.3 SWAM Practice 3: Integrity checking mechanisms are used to verify software, firmware, and information integrity

SWAM Practice 3 is built on ensuring that software assets are accurately managed and maintained through verification of their identity, both in the software inventory and in detection of software assets.

Table 15: Considerations Regarding the Integrity Checking of Software Assets

Considerations	Notations and Comments
How do we integrate our software authorization process with the update of software assets?	
How does the discovery of software vulnerabilities integrate with the software authorization process?	
How long does it take us to update the software authorization status after a known vulnerability is released?	
How long does it take us to remove a software asset once authorization is revoked or has expired?	
Do we have a centralized mechanism for uniquely identifying software assets?	

Considerations	Notations and Comments
Who in our organization has responsibility for the separate parts of software asset management (e.g., AV, baseline, network, and threat and vulnerability management)? How is the information among the different groups collected and coordinated?	
What technologies are used for software integrity checks?	
What tools does our organization have deployed that are listed on the BPA?	
What tools has our organization deployed that do not fully satisfy the SWAM service?	
Who validates the mechanism for uniquely identifying software?	
Have we defined a responsible party for resolving integrity violations?	
Do we have a technical control that isolates, restricts, controls, and/or reports on non-approved software?	
What do we currently do when unauthorized software assets are discovered on our network?	

5.2.3 Manage Configuration Settings (CSM) Security Capability

The CSM security capability provides a D/A visibility into risks associated with improper or non-compliant security-related configuration settings for authorized hardware and software. This security capability provides a method to ensure that all hardware assets are compliant with Common Configuration Enumeration (CCE) requirements. The CSM security capability is dependent on the existence of both an authorized hardware and authorized software inventory as developed for the HWAM and SWAM security capabilities.

The operational execution of CSM security capability is dependent on the following desired state specifications:

- All hardware and software assets are configured according to policy for all devices (i.e., hardware assets).
- All authorized hardware and software with security-related configuration settings have a configuration settings specification that defines the policy for that asset type.

Based on the CSM desired state specifications, the following practices support a D/A in successfully implementing the CSM security capability:

- CSM Practice 1: The development and testing environment(s) are separate from the production environment.
- CSM Practice 2: A baseline configuration of information technology/industrial control systems is created and maintained.
- CSM Practice 3: Configuration change control processes are in place.

5.2.3.1 CSM Practice 1: The development and testing environment(s) are separate from the production environment.

CSM Practice 1 is concerned with configuration management and organizational processes of a development and testing environment that is separate from the operational environment. Separating these two environments helps reduce risk by catching vulnerabilities before they are introduced into the live operational environment. It is key for the configuration settings of hardware and software in the development and testing environments to be synchronized with the operational environment, which means they need to be managed similarly, but separately. To assess the organization's ability to fully implement the three CSM practice**s**, there are considerations that provide insight on existing organizational policies and practices, which relate directly to CSM security capabilities.

Table 16: Considerations Regarding Configuration Management and Organizational Processes of the Development and Testing Environment(s)

Considerations	Notations and Comments
What change control roles and processes exist for our development and testing environment?	
What is the process for configuring software assets in our development and testing environment?	
What is the process for software configuration updates in our development and testing environment?	
Who has authority to approve software configuration settings in our development and testing environment?	
How often is the baseline configuration updated?	
How often do we validate and verify configurations?	
Do we need additional tools or have platform gaps in our development and testing environment?	
How do we manage software configuration in our development and testing environment (e.g., automated tools, spreadsheets, data storage, etc.)?	
Do we have policies that address configuration establishment, maintenance, and approval/authorization for software development and testing?	
Do we have policy gaps and if so, are they being addressed?	
What authoritative security-related configuration sources (e.g., USGCB, STIG, SCAP, Center for Internet Security benchmarks) are used in our development and testing environment?	
Are we using the latest version of the standard(s)?	
Do we have an exception management process (exceptions from the baseline) in our development and testing environment?	

Considerations	Notations and Comments
Do we use CCEs to uniquely identify configuration settings in our development and testing environment?	
How do we uniquely identify configurations in our development and testing environment?	
Are unique identifiers used (e.g., SWID tags)?	
Who manages approval/authorization and maintenance of baselines in our development and testing environment?	
What compliance mechanisms are used for assuring the adherence of baselines in our development and testing environment?	
What process, if any, is used to align baselines when they are found to be out of compliance? If none, what actions are taken?	
How do we validate the baseline in our development and testing environment?	
Do we have custom developed or integrated software baselines in our development and testing environment?	
Do we have configuration baselines and configuration checks integrated into the software management process in our development and testing environment?	

5.2.3.2 CSM Practice 2: A baseline configuration of information technology/industrial control systems is created and maintained.

CSM Practice 2 contains some overlapping considerations from Practice 1 but is intended to focus on baseline configurations of the operational environment versus the development and testing environment(s).

Table 17: Considerations Regarding Configuration Management of the Operational Environment

Considerations	Notations and Comments
What tools and platform gaps exist across our organization?	
How do we manage software configuration (e.g., automated tools, documents, data storage)?	
What authoritative security-related configuration sources do we use (e.g., USGCB, STIG, SCAP, CIS benchmarks)?	
Are we using the latest version of the standard(s)? If not, is there a defined reason (i.e., compatibility with current sensors)?	
How do we uniquely identify configurations settings across all hardware and software assets? Do we employ CCEs in the process?	
Are unique identifiers used (e.g., SWID tags)?	

Considerations	Notations and Comments
What baselines are used across our organization?	
Who manages the approval/authorization and maintenance of baselines?	
How do we validate the baseline?	
Do we have custom developed and integrated software baselines?	
What existing configuration management tools can be or will be integrated into the CDM collection system?	
What configuration attributes do we collect?	
Is time stamping used when we collect configuration status?	
If time stamping is used, how do we integrate time stamps into the detection and remediation process?	
What is the mean time it takes us to identify a misconfiguration?	
If there is a misconfiguration identified, what is the median time to repair it?	

5.2.3.3 CSM Practice 3: Configuration change control processes are in place.

CSM Practice 3 contains some overlapping considerations from Practice 1 but is intended to focus on organizational change control processes in the operational environment versus the development and testing environment(s).

Table 18: Considerations Regarding the Implementation of Change Control Processes

Considerations	Notations and Comments
What process do we use for software configuration updates?	
Who has authority to approve software configuration settings?	
How often do we update the baseline configuration?	
How often do we review configurations?	
What change control roles and processes exist for our organization?	
Do we have a process for configuring software assets?	
Do we have policies that address configuration establishment, maintenance, and approval/authorization?	
Are there any policy gaps across our organization?	
How are configuration baselines and configuration checks integrated into the policies, configuration management process, and software management process?	
Is there a coordination process between our organization and hardware and software owners?	

Considerations	Notations and Comments
Who in our organization has hardware and software settings approval authority?	
Who is the final authority on hardware and software setting authorization determinations?	
Do we have an exception management process (exceptions from the baseline)?	

5.2.4 Manage Vulnerabilities (VUL) Security Capability

The VUL security capability provides the D/A visibility into the known vulnerabilities present on its networks. Known vulnerabilities are those with a Common Vulnerability Enumeration (CVE) identifier or discovered by the local organization and associated with a specific set of software assets.

The operational execution of VUL security capability is dependent on the following desired state specifications:

- Software products (i.e., software assets) installed on all devices (i.e., hardware assets) are free of known vulnerabilities.
- The list of known vulnerabilities is up-to-date.

Based on these VUL desired state specifications, the following practices support a D/A in successfully implementing the VUL security capability:

- VUL Practice 1: Threat and vulnerability information is received from information-sharing forums and sources.
- VUL Practice 2: A vulnerability management plan is developed and implemented.
- VUL Practice 3: Vulnerability scans are performed.

5.2.4.1 VUL Practice 1: Threat and vulnerability information is received from information sharing forums and sources.

Table 19: Considerations Regarding Threat and Vulnerability Information

Considerations	Notations and Comments
Do we acquire vulnerability/threat information and data from external sources?	
How do we communicate threat and vulnerability information with selected external organizations and groups (e.g., US-CERT)?	
Do we communicate and maintain ongoing information sharing on threats and vulnerabilities with other components and D/As?	

5.2.4.2 VUL Practice 2: A vulnerability management plan is developed and implemented.

Table 20: Considerations Regarding an Organization's Vulnerability Management Program

Considerations	Notations and Comments
Do we have a software vulnerability management program?	
What tools and mechanisms do we use for vulnerability management?	
How current is our vulnerability management policy?	
What is the average time to make changes or updates to our vulnerability management policy?	
What office or point of contact is responsible for vulnerability management within our organization?	
How is vulnerability management performed across our organizational components and different network segments?	
Are vulnerability scans centrally managed?	
How is the vulnerability management program integrated with our change management process?	
Do we have a threat awareness program for the CDM scope?	
Is the threat awareness program centralized or federalized?	
Does our defined CDM scope include non-Windows-based software assets?	
How do we identify vulnerabilities in proprietary software assets?	
How do we identify non-Windows-based software vulnerabilities?	
Do we prioritize software assets beyond the Common Vulnerability Scoring System (CVSS)?	

5.2.4.3 VUL Practice 3: Vulnerability scans are performed.

Table 21: Considerations Related to How an Organization Conducts Vulnerability Scans

Considerations	Notations and Comments
Do we conduct credentialed vulnerability scans of assets?	
What percentage of our organizational assets is covered by credentialed vulnerability scans?	
How often are credentialed vulnerability scans performed?	
What actions are taken when vulnerabilities are discovered?	
What authoritative sources (e.g., CVE, CVSS) are used to identify vulnerabilities?	
Is the vulnerability management scope aligned with our authorized hardware and software inventories?	
How do we designate our vulnerability management scope?	
What is our process for conflict resolution within the vulnerability management program?	
What office is responsible for resolving conflicts within the vulnerability management program?	
How often do we scan assets for vulnerabilities?	
Do we have a defined process/policy for blacklisting a software asset with no known vulnerability mitigation?	
Do we scan for common software weaknesses using Common Weakness Enumeration (CWE) data?	

6 Conclusion

The responses to the considerations presented in this document help a D/A better understand how current operations can be transitioned or configured to prepare for CDM asset-based security capability implementation. Many existing processes and procedures executed by D/As today directly support CDM asset-based security capabilities or, with minor adjustments, can be integrated into required CDM asset-based processes. Compared to NIST CSF subcategories, D/As will be better prepared to implement CDM asset-based security capabilities by understanding how their current practices support each CDM asset-based security capability.

Comments can be sent to the CDM Program office at cdm.fnr@hq.dhs.gov.

7 Appendix A: References

Caralli, Richard A.; Allen, Julia H.; White, David W. (2010-11-24). *CERT Resilience Management Model (CERT-RMM): A Maturity Model for Managing Operational Resilience* (SEI Series in Software Engineering). Pearson Education.

8 Appendix B: Program Resources

- US-CERT CDM: http://www.us-cert.gov/cdm
- DHS CDM: http://www.dhs.gov/cdm
- Homeland Security Information Network (HSIN): https://hsin.dhs.gov/dhs/CDM/Pages/ (This website contains training and portal pages for technical working groups. For access and other information, see contract staff.)
- CDM Program contact information: cdm.fnr@hq.dhs.gov, GSA CDM www.gsa.gov/cdm, or contact cdm@gsa.gov