National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Interagency Report 7670
(Draft)

Proposed Open Specifications for an Enterprise Remediation Automation Framework (Draft)

David Waltermire
Christopher Johnson
Matthew Kerr
Matthew Wojcik
John Wunder

14

15 **NIST Interagency Report 7670 (Draft)**

Proposed Open Specifications for an Enterprise Remediation Automation Framework (Draft)

David Waltermire
Christopher Johnson
Matthew Kerr
Matthew Wojcik
John Wunder

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

February 2011

U.S. Department of Commerce

Gary Locke, Secretary

National Institute of Standards and Technology

Dr. Patrick D. Gallagher, Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Interagency Report discusses ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Interagency Report 7670 (Draft)
17 pages (Feb. 2011)

Acknowledgments

The authors wish to tank their colleagues who reviewed drafts of this document and contributed to its technical content. The authors would like to acknowledge John Banghart of NIST, Paul Cichonski of Booz Allen Hamilton, and Karen Scarfone of G2, Inc. for their insights and support throughout the development of the document.

Abstract

The success of SCAP in automated system assessment has fostered research related to the development of similar open specifications in support of enterprise remediation. Enterprise remediation is focused on delivering capabilities that allow organizations to identify, describe and implement desired system changes across the enterprise. Remediation actions can include changes to the configuration of an operating system or application, installation of a software patch, or the installation or removal of applications and libraries. This report examines technical use cases for enterprise remediation, identifies high-level requirements for these use cases, and proposes a set of emerging specifications that satisfy those requirements.

This report is a product of ongoing collaboration between the National Institute of Standards and Technology (NIST), the US Department of Defense, and the MITRE Corporation. Participation from a broader community of interested parties is actively sought to help define, refine and mature proposed remediation standards.

Audience

The primary audience of this paper is government and industry security analysts, security product developers, and operating system and application vendors. NIST welcomes feedback from these groups as well as members of the broader community of interest.

Table of Contents

List of Figures

List of Tables

1. Introduction

In recent years, automated information security assessment for the enterprise has been advanced through the widespread adoption of the Security Content Automation Protocol (SCAP), a suite of specifications that standardize the format and nomenclature by which security software products communicate software flaw and security configuration information. The SCAP component specifications have allowed enterprises to define security policy, monitor system state, perform software inventory, and evaluate system vulnerability and patch status. Further, because these are open specifications, organizations are not locked into single-vendor proprietary solutions for automated assessment, but instead can select tools from a wide range of vendors.

The success of SCAP in automated system assessment has fostered research related to the development of similar open specifications in support of enterprise remediation use cases. Within this paper, a remediation is defined as "a security-related[1] set of actions that results in a change to a computer's[2] state" and may consist of changes motivated by the need to enforce organizational security policies, address discovered vulnerabilities, or correct misconfigurations. Remediations can include changes to operating system and application software configuration settings, the installation of patches, and the installation or removal of applications, software components or libraries.

A vulnerability is an error, flaw, or mistake in computer software that permits or causes an unintended behavior or side effect to occur. Such behaviors may allow an attacker to:

- Execute commands as another user
- Access or modify data that is contrary to the specified access restrictions for that data
- Pose as another entity (e.g., user, organization, host)
- Affect the availability of a system resource

Common Vulnerabilities and Exposures (CVE) is the specified convention for naming known vulnerabilities within SCAP. CVE is utilized within this framework to correlate vulnerabilities with specific remediations.

A misconfiguration is a configuration setting that violates organizational security policies, introduces a possible security weakness in a system, or permits or causes unintended behavior that may impact the security posture of a system. These misconfigurations may include:

- Unauthorized services are found to be running
- Improper access control settings are detected
- Inadequate logging and auditing
- Encryption requirements are not enforced

Common Configuration Enumeration (CCE) is the specified convention for identifying and expressing configuration settings within SCAP. CCE is utilized within this framework to correlate vulnerabilities with specific remediations.

There are currently no existing open specifications for remediation analogous to the current SCAP assessment specifications. In the absence of open remediation specifications, integrating components

[1] It is understood that many of the technical use cases described in this paper in the context of system security also apply to general system change management, which is not necessarily motivated entirely by security concerns. Similarly, the proposed solutions outlined here may also have broader application. However, the scope of this effort is currently focused on security-relevant remediation activities.

[2] The proposed specifications may also be applicable to other types of IT assets, such as network devices (routers, firewalls, etc.), but the scope of this effort is currently focused on desktops, laptops, workstations and servers.

from different vendors to perform enterprise-wide remediation actions can be difficult, expensive, or even impossible. This lack of interoperability hampers many organizations' attempts to deploy comprehensive assessment and remediation capabilities. This report examines technical use cases for enterprise remediation, identifies high-level requirements for these use cases, and proposes a set of emerging specifications that address those requirements.

1.1 Technical Use Cases

The following technical use cases are a set of motivating scenarios for the development of open specifications in support of enterprise remediation capabilities:

- Use Case 1 – *Assess then remediate all*: Remediate one or more computing assets for all vulnerabilities and misconfigurations discovered during a prior assessment
- Use Case 2 – *Assess then selectively remediate*: Remediate one or more computing assets for a subset of vulnerabilities and misconfigurations discovered during a prior assessment
- Use Case 3 – *Independent remediation*: Apply one or more remediations to one or more computing assets irrespective of any prior assessment activities. This is not to say that certain pre-conditions may need to be evaluated before performing the remedy. For example, ensuring that the architecture is 64-bit before installing the 64-bit version of an application.

1.2 Remediation Workflow Components

The technical use cases introduced in Section 1.1 arise from enterprise remediation decision-making processes and their associated workflows. The key components of an enterprise remediation workflow are described in Table 1.

157 **Table 1. Remediation Workflow Components**

Component	Description
Remediation Policy Source	Public or private repository for remediation policy documents.
Remediation Policy	Set of remediation policy directives for computing assets. These directives may specify target platforms, parameter values, and a reference to a common remediation identifier. Remediation policies may define configuration settings that are to be applied, vulnerabilities to be remedied, and patches that must be applied. Such policies can be established at the enterprise level and may be tailored to meet the local operational needs of organizational elements or business units.
Remediation Management Tool	Tool responsible for evaluating assessment results, remediation policy, and remediation details to produce specific remediation tasking instructions for remediation tools.
Remediation Data Source	Public or private repository for detailed remediation information.
Remediation Tool	Tool responsible for applying individual remediations to specified assets.
Remediation Details	Publicly or privately held data that identifies the vulnerability or misconfiguration a remediation addresses, any prerequisites for performing the remediation and post-application instructions.
Assessment Results	Describes the vulnerabilities or misconfigurations discovered by an assessment or scanning tool and the metadata regarding how and when the assessment was performed (e.g., date & time of the scan, tool used, scan operator).
Remediation Tasks	Remediation instructions specifying which remediations are to be applied, when they are to be applied, and under what conditions.
Remediation Results	The outcome of attempted remediation tasks on particular assets.

158
159 The diagram shown in Figure 1 depicts the tools, interfaces and data exchanges in a notional enterprise
160 remediation workflow. Note that the Assessment, Remediation Management, and Remediation Tools
161 depicted in Figure **1** may be implemented as modules in an integrated product suite or as separate
162 applications, possibly from different vendors.
163
164

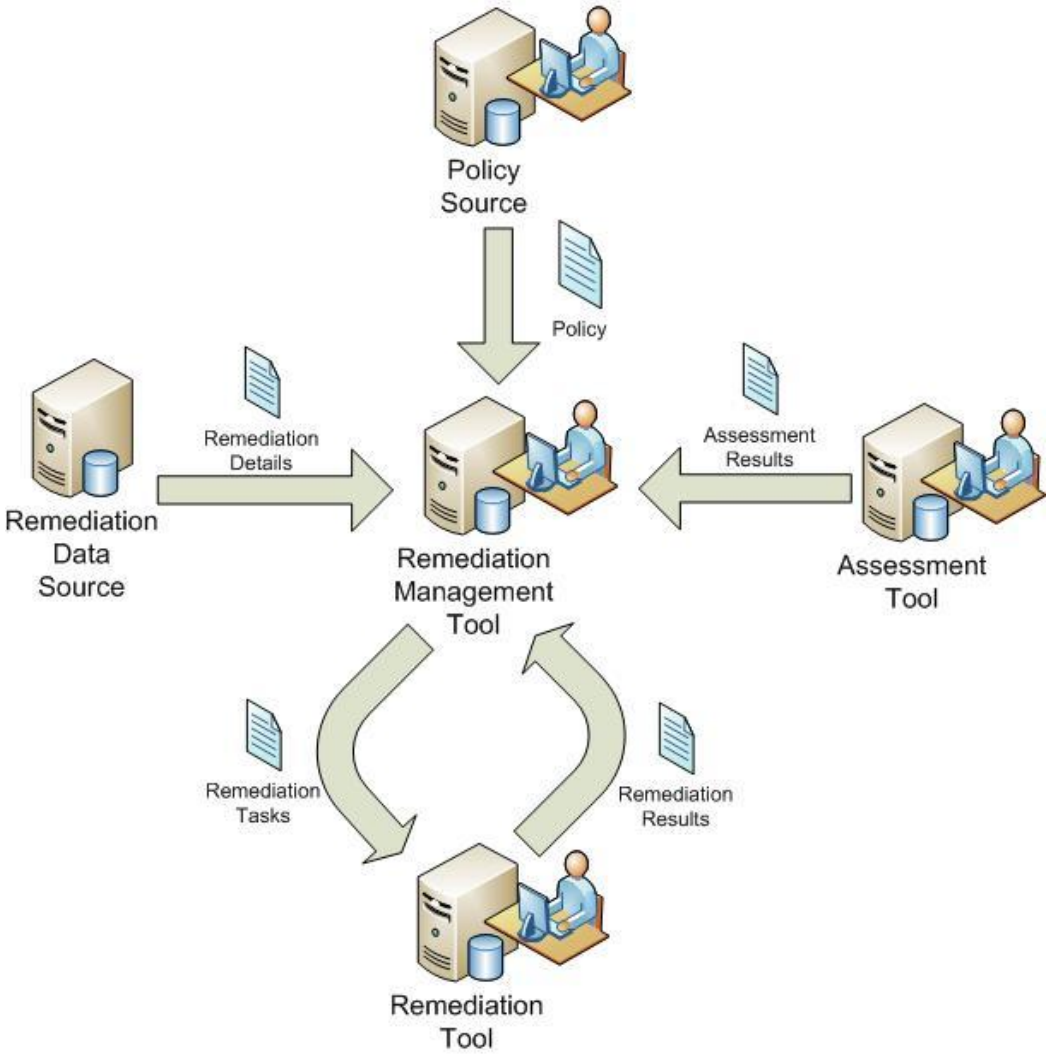

Figure 1: Enterprise Remediation Workflow Diagram

1.3 Derived Requirements (DR)

Based on the technical use cases identified in Section 1.1 and the enterprise remediation workflow depicted in Figure 1, the following high-level requirements were identified:

- DR1. Method for uniquely identifying a remediation
- DR2. Definition of an exchange format for basic remediation information
- DR3. Definition of additional data about a remediation, including mappings to applicable platforms, related vulnerabilities, or configuration issues
- DR4. Definition of a language for the exchange of the additional remediation data identified in DR3
- DR5. Method for specifying remediations for classes of assets
- DR6. Method for applying remediations to specific assets in an enterprise environment
- DR7. Method for reporting the results of an attempted remediation
- DR8. Method for expressing how to perform a remediation in a precise, machine-readable fashion

The remainder of this report proposes solutions intended to address these derived requirements.

2. Derived Requirements Details and Specifications

This section expands upon the derived requirements discussed in Section 1.3 by describing the scope, context, and purpose of these requirements. Additionally, this section identifies the minimal functional capabilities and data requirements necessary to satisfy each of the technical use cases.

2.1 Common Remediation Enumeration (DR1)

Common Remediation Enumeration (CRE) is the proposed name for a standardized list of identifiable remediations. CRE is the first emerging specification proposed in response to the currently understood technical use cases for enterprise remediation.

The scope of a CRE entry is the set of actions that must be taken to accomplish a distinct remediation objective (e.g., installing a software patch or changing the system configuration). As such, a single CRE could require that multiple atomic actions, such as changing a configuration value and installing a patch, be performed to achieve the desired end state.

A CRE entry consists of only the minimum amount of data required to differentiate one remediation from another:

- **Unique Identifier** - textual ID for the specific remediation being referred to. Because there is a need to enumerate organization-specific remediations in addition to those universally recognized, CRE will accommodate local identifiers. For example, an organization may choose to issue local CRE identifiers for internal, custom applications or for remediation actions that are specific to their operational environment. The CRE ID will contain a namespace component that identifies the organization that issued and controls the CRE entry. The remainder of a CRE ID is a non-semantic unique ID; it does not convey or encode any information about the remediation or impart any meaning.

- **Description** - brief paragraph intended for a human audience. The description, in conjunction with the supporting references, must provide sufficient information to allow a person to differentiate one remediation from another. The description is not intended to convey the details of the remediation actions, but only a concise description.

- **Supporting References** - links to authoritative sources where the remediation has been described (e.g., configuration guides, vendor security bulletins, patches). The references may provide additional supporting information about the CRE, including why it was created, how it is distinct from other similar CREs or additional technical discussions regarding the remediation.

- **Metadata** - Information about the CRE entries themselves will also be maintained, such as creation and modification dates, deprecation status, version information, and provenance.

CRE will foster interoperability by supporting the standardized exchange of remediation-related content across organizations and by enabling the coordination of IT security actions across a variety tools. CRE can be used in much the same way as CVE and CCE are used today in support of vulnerability and configuration management activities respectively. CRE identifiers will be used throughout enterprise remediation workflows; acting as the primary key in the specification of remediation policy, enabling the retrieval of detailed remediation information, identifying desired remediation actions during tasking, and conveying the results of attempted remediations.

CRE describes the data that is required to support the technical use cases identified; it does not prescribe a database format, schema or presentation model. The CRE data exchange format described in Section 2.2

presents a proposed lightweight transport format for the exchange of CRE information. CRE will be more fully described in a forthcoming specification.

2.2 CRE Data Exchange Format (DR2)

An exchange format for CRE entries and related metadata (as described above) is required to enable the transfer of CREs between parties and tools. This transport format allows the exchange of either the standard CRE list or organization-specific CREs. The CRE data exchange format is envisioned as a lightweight, XML-based schema that serves as the standard import, export, and exchange format for basic remediation information as provided by CRE.

The CRE data exchange format will be described in a forthcoming specification.

2.3 Extended Remediation Information (DR3)

CRE provides a core set of basic remediation information. Supplemental remediation information is required in order to meet the described use cases. This related information, though not part of the CRE entry proper, describes the entry more fully, including describing relationships to other key concepts.

As CRE is analogous to CVE, so is Extended Remediation Information (ERI) analogous to the additional CVE-related information available in the National Vulnerability Database (NVD). NVD provides mappings of CVEs to weakness types and affected software products, impact metrics, and other information that complements the information present in the base CVE entry.

Extended Remediation Information defines additional information about CRE entries necessary to fully support enterprise remediation workflows. While a sizeable collection of remediation information exists today, it lacks structural consistency, varies in completeness from vendor to vendor, and often must be retrieved from multiple sources. By specifying desired ERI, providers of remediation information have a template that describes the desired content.

ERI may describe:

- Applicable platforms (i.e., CPEs) for the remediation
- Vulnerabilities (i.e., CVEs) that a remediation is intended to resolve
- Misconfigurations (i.e., CCEs) that a remediation is intended to resolve
- Human- or machine-readable prerequisites for remediation (e.g., other remediations)
- Descriptions of remediation actions (human- or machine-readable)
- Required actions on success or failure of an attempt to apply the remediation (human- or machine-readable)

ERI does not prescribe a database format or schema or any other presentation model. It simply identifies the additional data that may be required to support the identified technical use cases, beyond the base CRE entries. The ERI data exchange format described in Section 2.4 presents a proposed lightweight transport format for the exchange of ERI information.

ERI as described provides the information necessary to decide which remediations to include in an enterprise remediation policy, or to facilitate the selection of appropriate remediations to apply based on assessment results.

The ability to fully support the breadth of identified use cases, enabling maximum automation and tool integration, requires that ERI for all critical remediations be managed and maintained by some centralized authority or authorities.

ERI will be fully described in a forthcoming specification.

2.4 Extended Remediation Information Data Exchange Format (DR4)

A common representation of ERI is required to facilitate data exchange and to foster tool interoperability. The Extended Remediation Information data exchange format is proposed as a means of enabling efficient interchange of ERI data.

While ERI defines the remediation data necessary to support the described use cases, the data exchange format specifies a standardized format for the automated exchange of ERI between remediation information sources and remediation tools. ERI may also appear in machine-readable remediation policy documents.

The ERI data exchange format is envisioned as an XML-based schema that extends the CRE schema, allowing ERI documents to refer to the CRE entries they extend by CRE ID alone, or to contain the full contents of the CRE entry.

The ERI data exchange format will be fully described in a forthcoming specification document.

2.5 Remediation Policy Specification (DR5)

The Remediation Policy Specification defines how to associate particular remediations with various classes or types of IT assets. Such a capability allows organizations to specify allowed, preferred, or required remediations for specified collections of IT assets.

Those asset types may be defined by:

- Platform type (e.g., desktop, notebook, server)
- Software inventory (i.e., presence of a particular product)
- Presence of specific vulnerabilities
- Current configuration of the IT asset
- Functional categories (e.g., web server, database server)
- Organizational boundaries
- Combinations of the above

The Remediation Policy Specification provides a standard format that enables an organization to constrain the full set of *possible* remediation options for a given circumstance to a smaller *allowed* subset. For example, suppose there are two known CRE entries for a particular vulnerability, one identifying a patch and the other a mitigating workaround. An organization's remediation policy might indicate that in most cases, the patch should be installed, but in cases where a third-party application with known conflicts with the patch is also present, the workaround should be applied instead.

A remediation policy in effect conveys remediation decisions that have been made in advance, simplifying the decisions that must be made synchronously in a remediation workflow. In cases where the remediation policy specifies a single remediation for a given situation, full automation of remediation action may be possible. The Remediation Policy Specification defines how remediation policies may be expressed and exchanged in an open, unambiguous, and machine-readable format.

Initial discussion of the requirements for the Remediation Policy Specification suggests XCCDF could potentially be used for this purpose, either in its current form or with some modifications. The use of XCCDF as this expression will be investigated, as will other viable alternatives.

The Remediation Policy Specification will be fully described in a forthcoming specification document.

2.6 Remediation Tasking Language (DR6)

In contrast to the Remediation Policy Specification, which assigns remediations to classes of assets, the proposed Remediation Tasking Language (RTL) provides a standardized format to direct compliant tools to enact specific remediations on specific assets. RTL documents represent the output of the remediation decision process, and function as a standardized input format for remediation tools.

Remediation Tasking Language documents specify:

- Which assets to remediate
- Which remediation actions to perform
- What values are to be used in performing each remediation (e.g., number of characters to set as the minimum password length)

Other operational parameters, such as deferral options, may also be included.

Development of the Remediation Tasking Language will take into consideration other emerging reporting and control specifications being considered in the overall security automation architecture. This evaluation will include assessing conceptual alignment and the potential for schema reuse.

The Remediation Tasking Language will be fully described in a forthcoming specification document.

2.7 Remediation Results (DR7)

In order to determine what follow-up steps, if any, are necessary, the results of a remediation attempt must be communicated back to the tool or process that requested the remediation. These Remediation Results convey the outcome (e.g., success/failure/error) of attempted remediation actions as reported by the remediation tool. Remediation Results also enable roll-up reporting and provide enhanced situational awareness.

These results include, by asset:

- Outcome of the attempted remediation
- Explanatory information, when the remediation attempt was unsuccessful
- Date and time the remediation was performed
- Date and time the remediation is scheduled to be performed, if deferred
- Initiator of the deferral action

Remediation Results are not intended to serve as an authoritative assertion of whether an asset is still subject to a vulnerability or misconfiguration that a remediation was intended to address. Initiating a reassessment of the affected asset using the appropriate assessment tool is the preferred method for making such a determination. Remediation Results are most ideally suited for supporting follow-on decisions in the remediation workflow, such as whether to attempt a failed remediation again, whether to override the deferral of a remediation by a user, or as decision support material in determining the need for further assessment.

Development of the Remediation Results will take into consideration other emerging reporting formats being considered in the overall security automation architecture. This evaluation will include assessing conceptual alignment and the potential for schema reuse.

Remediation Results will be fully described in a forthcoming specification document.

2.8 Open Vulnerability Remediation Language (DR8)

The Open Vulnerability Remediation Language (OVRL) is intended to provide the capability to express the low-level, machine-readable instructions necessary to perform a remediation. An OVRL statement is directly interpretable by a compliant remediation tool, allowing the tool to carry out the remediation. As CRE is similar to CVE or CCE, OVRL is similar to OVAL.

An OVRL statement would express, in machine-readable form:

- Prerequisites for successful remediation
- Manifest of changes to be made to the system, including ordering of these operations
- Follow-up actions (e.g., reboot, policy refresh, service restart)
- Error-handling instructions

OVRL provides transparency into the remediation process and allows remediations to be precisely and unambiguously defined. Enterprises using OVRL-based remediation tools are afforded greater visibility and control of the low-level remediation actions being performed. This may, in some cases, reduce the need for mapping activities around CRE, as OVRL-compatible tools simply consume the OVRL statements and follow the prescribed steps. "Zero-day" remediations or customized remediations can be enacted with minimal coordination delays, as tool vendors are not required to map CREs to proprietary remediation actions. OVRL statements are expected to use CRE IDs as the primary identifier of the remediations they more fully describe.

OVRL will be fully described in a forthcoming specification document.

3. Architecture and Data Flows

This section describes how the capabilities discussed in the derived requirements in Section 2 of this document are employed within the data flows of a notional enterprise remediation architecture. These emerging remediation capabilities are designed to work in concert with existing scanning capabilities to allow orchestration of remediation activities within the enterprise.

Figure 2 depicts each of the proposed remediation data flows as they might be employed within an enterprise remediation workflow. A derived requirement number is used to identify the use of the proposed specifications within the data flows.

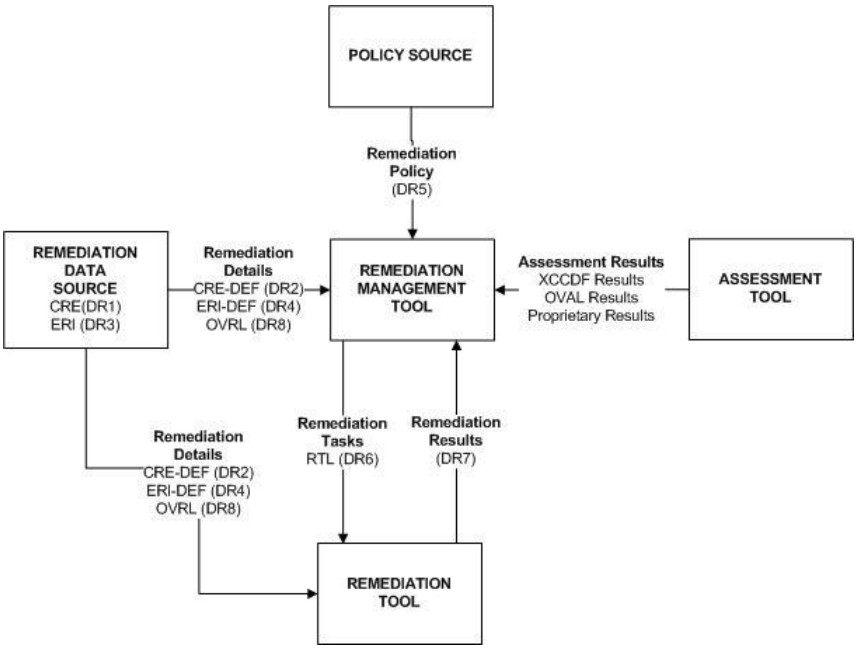

Figure 2: Enterprise Remediation Data Flow Diagram

Table 2 below describes the proposed remediation data flows including source and destination of the data, data flow contents and their associated derived requirement numbers.

413 **Table 2. Enterprise Remediation Data Flow Description**

414

Data Flow Name	Data Flow Description
Remediation Policy (DR5)	This data flow originates from a Remediation Policy Source and is sent to the Remediation Management Tool. It contains security policy directives for information technology systems expressed using the common, open, remediation policy language described in *Derived Requirement 5*.
Remediation Details (DR2) (DR4) (DR8)	This data flow originates from a Remediation Data Source and is sent to the Remediation Management Tool and the Remediation Tool. It contains the detailed remediation data required to formulate remediation instructions and to perform endpoint remediation actions. This data flow includes remediation identifiers, extended remediation information and low-level remediation instructions expressed using the formats defined in the specifications identified in *Derived Requirements 2, 4, and 8*.
Assessment Results	This data flow originates from a security Assessment Tool and is sent to the Remediation Management Tool. It contains detailed assessment results from information technology assets and identifies settings that do not comply with the organizations security policy and are candidates for remediation. This data flow includes assessment results expressed using the SCAP component specifications[3] including: XCCDF, OVAL and OCIL.
Remediation Tasks (DR6)	This data flow originates from the Remediation Management Tool and is sent to the Remediation Tool. It contains the remediation tasking instructions required for remediation tools identifying target assets, remediation actions and values as defined by the specification identified in *Derived Requirement 6*.
Remediation Results (DR7)	This data flow originates from the Remediation Tool and is sent to the Remediation Management Tool. It contains the results of the remediation actions attempted by the Remediation Tool expressed in the common format defined in the specification identified in *Derived Requirement 7*.

[3] For more information on SCAP components refer to the NIST SP 800-126r1: http://csrc.nist.gov/publications/PubsSPs.html#800-126-r1

4. Appendix A—Acronyms and Abbreviations

Selected acronyms and abbreviations used in the report are defined below.

CCE	Common Configuration Enumeration
CPE	Common Platform Enumeration
CRE	Common Remediation Enumeration
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
ERI	Extended Remediation Information
IR	Interagency Report
IT	Information Technology
ITL	Information Technology Laboratory
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
OVAL	Open Vulnerability and Assessment Language
OVRL	Open Vulnerability Remediation Language
RTL	Remediation Tasking Language
SCAP	Security Content Automation Protocol
SP	Special Publication
XCCDF	eXtensible Configuration Checklist Description Format