Enterprise Remediation Automation

IT Security Automation Conference
September 27-29, 2010

Chris Johnson
Computer Security Division
Information Technology Laboratory
The National Institute of Standards and Technology (NIST)

Agenda

- Background
- Definitions
- Remediation Components
- Use Cases
- Remediation Workflow
- Proposed Specifications

Goal

Produce standardized security automation capabilities that impart greater efficiency in enterprise remediation processes

Approach

Explore the technical use cases for remediation and identify opportunities to enhance existing remediation capabilities and foster innovation through standardization.

Result

A proposed suite of 8 specifications that describe naming conventions, data exchange formats, and languages for remediation

Remediation

A set of actions that results in a change to the state of an IT asset that may be motivated by the need to enforce organizational security policies, address discovered vulnerabilities, or to correct an improper/insecure system configuration setting

Enterprise Remediation

Describes remediation capabilities that span an organization and address the:

- Definition, application and enforcement of organizational security remediation policies
- Management of remediation tasks
- Dissemination of remediation instructions
- Reporting the results of remediation attempts

Components of Automated Enterprise Remediation

Collection of individually maintained, community developed, open specifications that can be used to identify, describe and implement system changes across the enterprise

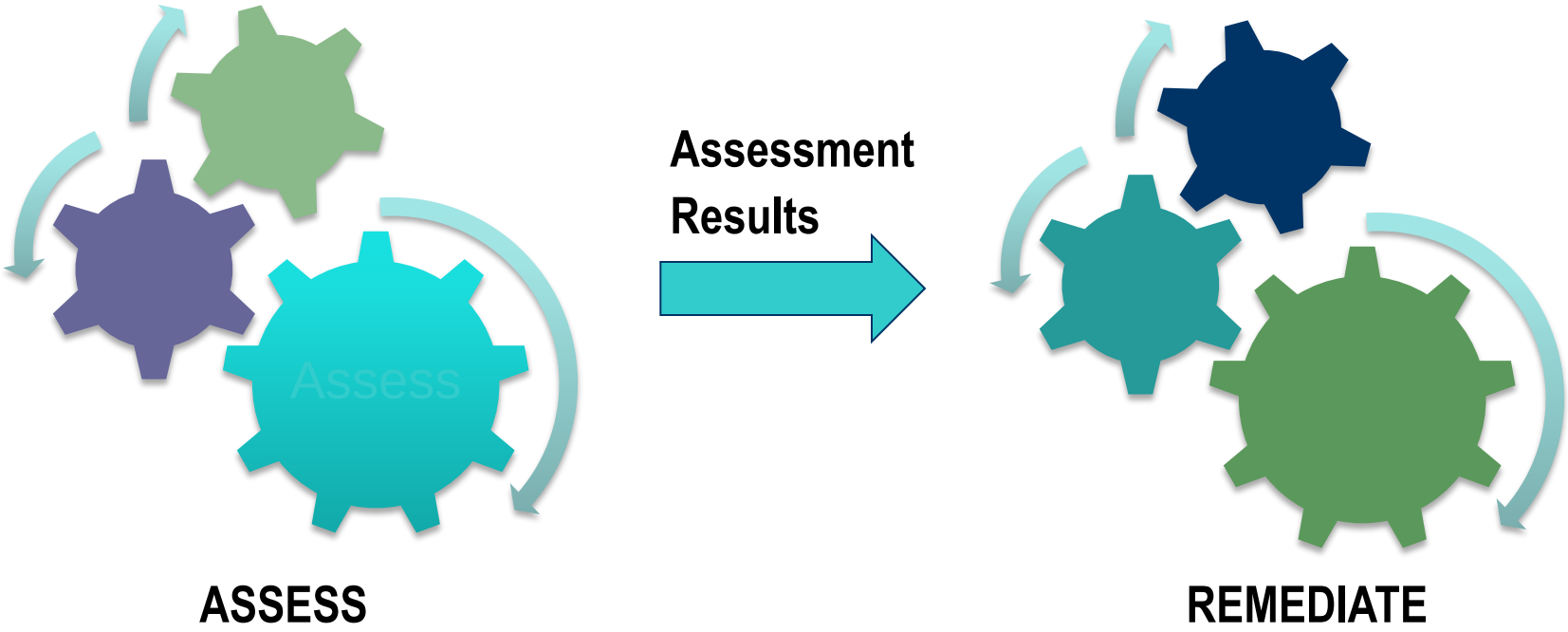
- Component specifications that establish conventions for identifying, describing, tasking and performing remediation actions
- High-level specifications define how the component specifications are used in concert to deliver capabilities to the security automation community

Body of reference data expressed in accordance with the specifications

Exploring Some Use Cases

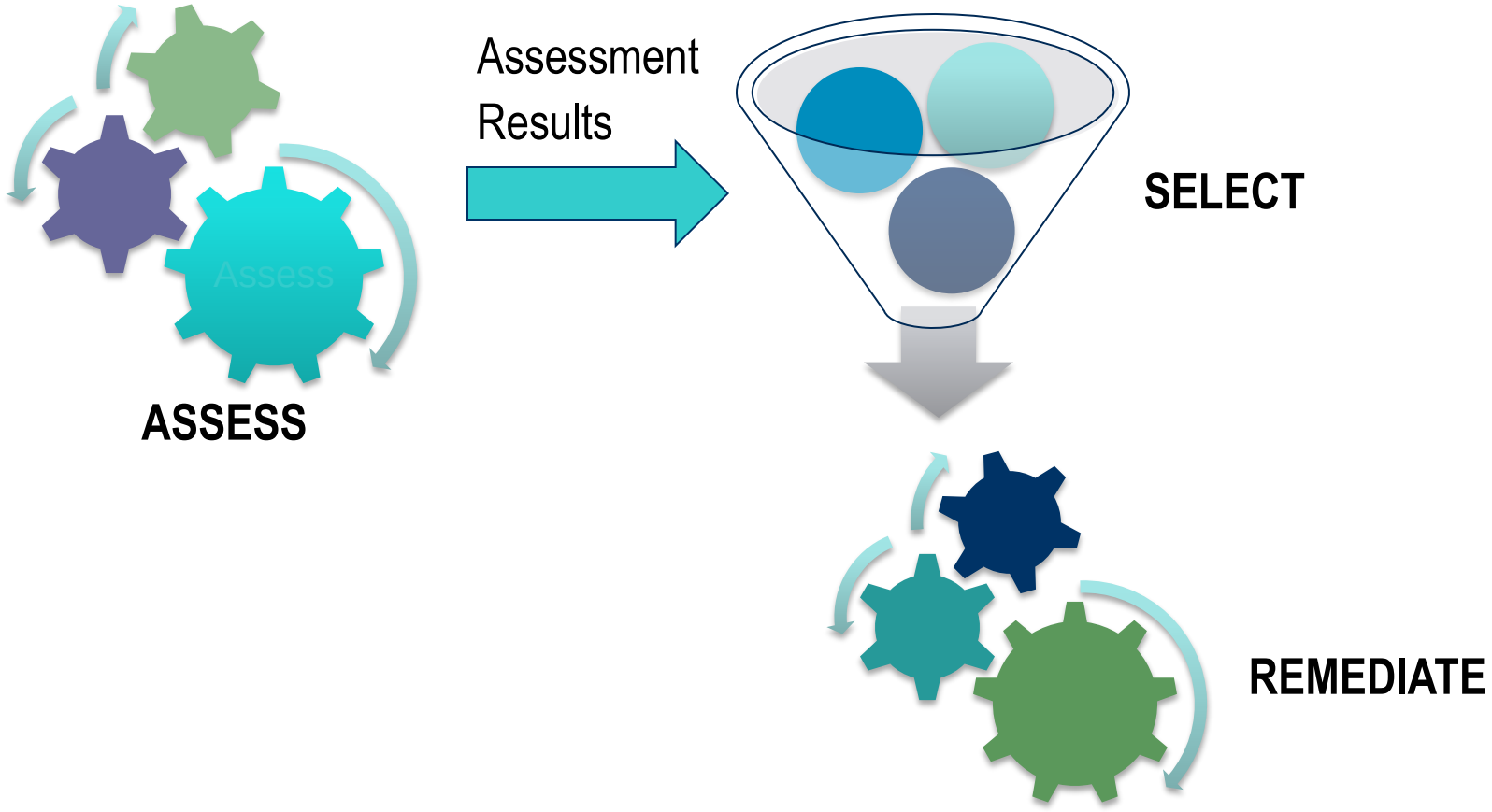
Use Case 1
Comprehensive Remediation

GOAL: Remediate one or more computing assets for all vulnerabilities and misconfigurations discovered during a prior assessment



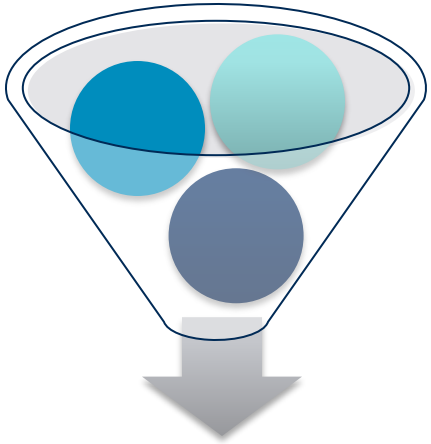
Use Case 2
Selective Remediation

GOAL: Remediate one or more computing assets for a subset of vulnerabilities and misconfigurations discovered during a prior assessment

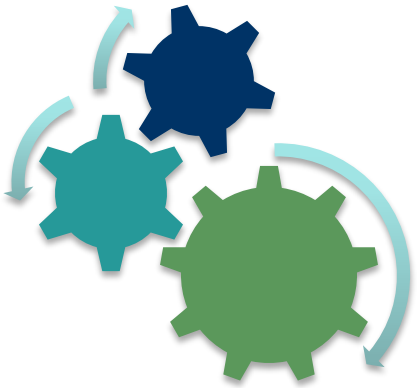


Use Case 3
Independent Remediation

Goal: Apply one or more remediations to one or more computing assets regardless of their current security state (as determined by an assessment scanner).



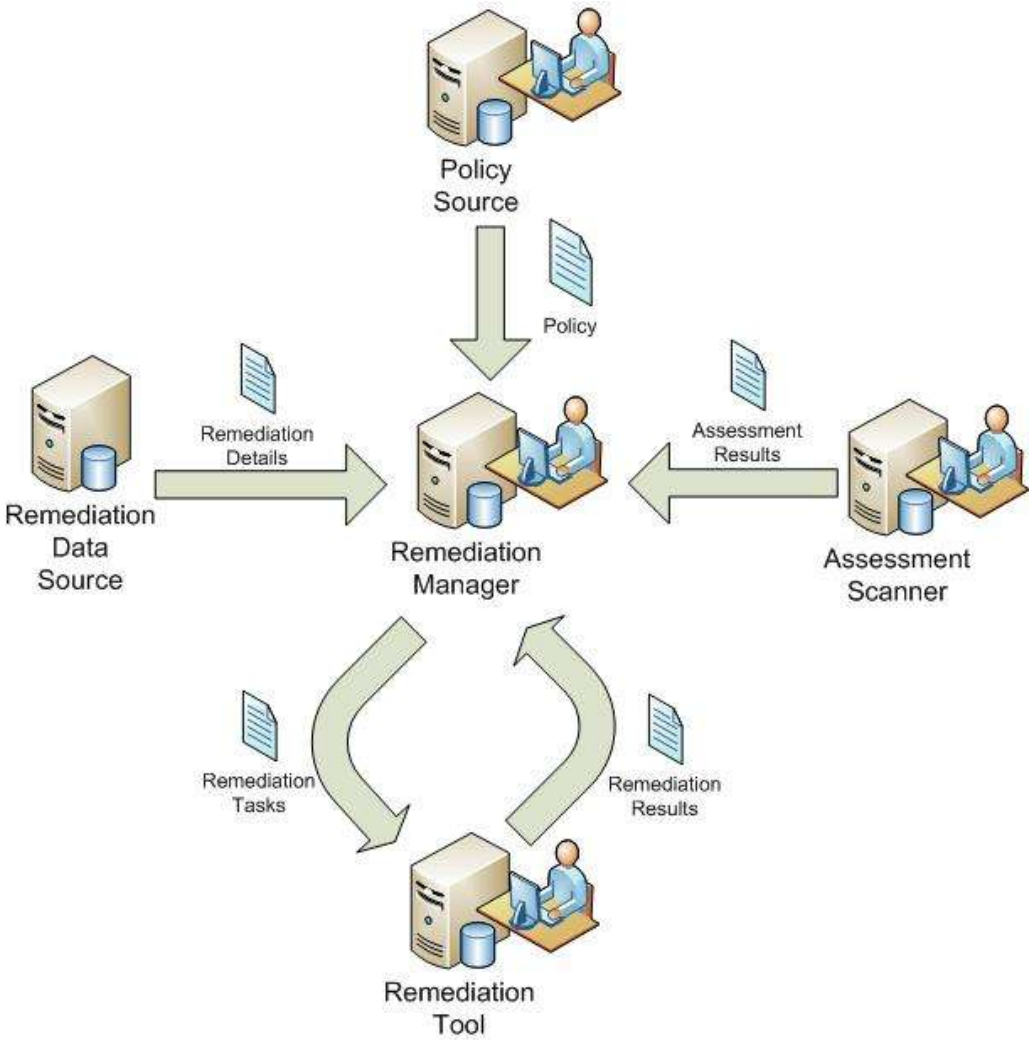
SELECT



REMEDIATE

Enterprise Remediation Logical Workflow Diagram

Remediation Information could originate from a product vendor, security tool database, third-party source, or it may reside in a local repository within the organization



Overview
Proposed Specifications (1)

Common Remediation Enumeration (CRE)

- Common names and basic remediation information

CRE Data Exchange Format

- Exchange format for CRE content

Extended Remediation Information (ERI)

- Mappings and other supplemental remediation details

ERI Data Exchange Format

- Exchange format for ERI content

Overview
Proposed Specifications (2)

Remediation Policy

- Express remediation policy

Remediation Tasking Language

- Ability to issue remediation directives to tools

Remediation Results Format

- Common format for the outcome of remediation attempts

Open Vulnerability Remediation Language (OVRL)

- Language for constructing machine-readable instructions necessary to perform the desired remediation

Common Remediation Enumeration (CRE)

- Assign a common identifier to the set of actions that must be performed to accomplish a distinct remediation objective
- CRE entry will contain the minimum amount of information necessary to distinguish it from other CRE entries and to describe its purpose

CRE Data Fields

- Unique Identifier
- Prose description of the remediation
- Conceptual Parameters
- Supporting References
- Metadata
  - Creation/Modification Dates
  - Entry Status
  - Version
  - Provenance

CRE Use Cases
Initial Configuration

Goal: Initial configuration of a system(s) to be in compliance with a predefined policy

Success Scenario: User of a standards-based remediation tool dispatches a series of remediation tasks to bring the target system(s) into compliance. The remediation actions are identified using CREs. Remediation tool performs the remediation tasks associated with the selected CREs and issues a report indicating that the CREs were successfully performed.

Failure Condition: Remediation tool is unable to complete all the assigned remediation tasks and issues a report that identifies CREs that were successfully applied and those that failed with an accompanying error message.

CRE Use Cases
Vulnerability Remediation

Goal: Remedy software flaws (CVE) detected by an assessment scanner

Success Scenario: User of a standards-based remediation tool selects and dispatches the appropriate CRE remediation actions for the CVEs detected. The remediation tool performs the selected remediation tasks and issues a report indicating that the CREs that were successfully applied.

Failure Condition: Remediation tool is unable to complete the assigned remediation tasks and issues a report that identifies the CREs that were not successfully applied and an accompanying error message.

CRE Use Cases
Compliance Enforcement

Goal: Remedy a non-compliant configuration setting (CCE) detected by an assessment scanner

Success Scenario: User of a standards-based remediation tool selects and dispatches the appropriate CRE remediation action for the CCE detected. The remediation tool performs the selected remediation task and issues a report indicating that the CRE was successfully applied.

Failure Condition: Remediation tool is unable to complete the assigned remediation task and issues a report that identifies the CRE that was not successfully applied and an accompanying error message.

CRE Sample Entry

Name	Value
CRE ID	cre:org.sample.cre.draft:1
CRE Description	Set the desired permissions on file sample.exe
Parameters	Desired file access permissions
Platform	cpe:/o:vendor_test:sample_os
References	http://www.sample.org/security/guidance
Entry Created	12 October 2009
Entry Modified	07 July 2010
Entry Version	2
Submitter	A3Q
Deprecated	FALSE

An Additional Note on CREs

A desired end state can often be reached in multiple ways - for example, a service may be disabled by:

- Commenting out the service startup command in a configuration file
- Changing the file permissions on the executable associated with the service
- Removal of the executable associated with the service

A separate CRE would be issued for each of these examples because the method and parameters for implementing the change are unique.

Extended Remediation Information (ERI)

- ERI captures additional information related to CRE entries – information that is often needed to fully support the enterprise remediation use cases described
- Capturing this supplemental data in ERI allows CRE to be much more lightweight and stable
- This approach is analogous to CVE (which carries essential identifying information about the vulnerability) and the extended data is available through the National Vulnerability Database (NVD) and its vulnerability data feeds.

ERI Data Fields

- Unique Identifier
- CRE Reference
- Indicators
- Parameter Mappings
- Supersedes
- Prerequisites
- Operational Impact
- Reboot
- Additional Metadata

ERI Use Cases
CRE Discovery based on CVE/CCE

Goal: Identify CREs that are relevant to a particular CVE (software flaw) or CCE (configuration setting)

Scenario: User submits a CVE or CCE identifier to an ERI repository as part of a query and is presented a list of candidate CREs. If the ERI repository is unable to locate an appropriate CRE for the CVE or CCE submitted an informational message is displayed.

ERI Use Cases
CRE Discovery based on CPE

Goal: Identify CREs that are relevant to a particular CPE (hardware, operating system or application)

Scenario: User submits a CPE identifier to an ERI repository as part of a query and is presented a list of candidate CREs. If the ERI repository is unable to locate an appropriate CRE for the CPE submitted an informational message is displayed.

ERI Use Cases
Operational Impact

Goal: Determine if a particular CRE has any reported operational impacts.

Scenario: User submits a CRE identifier to an ERI repository as part of a query and is presented information describing the possible operational impacts related to the CRE. If the ERI repository is unable to locate any information regarding operational impact for the CRE submitted an informational message is displayed.

ERI Use Cases
Reboot Requirements

Goal: Identify all CREs that require a reboot of the system for a particular CPE.

Scenario: User submits a CPE identifier to an ERI repository as part of a query and is presented a list of CREs for which a system reboot is required. If the ERI repository is unable to locate any CRE meeting the criteria an informational message is displayed.

ERI Sample Entry

Name	Value
ERI ID	eri:org.sample.eri.draft:101
CRE Reference	cre:org.sample.cre.draft:27
Indicators	CCE-2824-1
Parameter Mappings	Conceptual Value 1:Enable Literal Value 1:1 Conceptual Value 2:Disable Literal Value 2:0
Supersedes	None
Prerequisites	cre:org.sample.cre.draft:9
Operational Impact	None
Reboot	TRUE
Submitter	A3Q
Deprecated	FALSE

Remediation Policy Specification

- Allows organizations to specify allowed, preferred, or required remediation actions for specified collections of IT assets.
- Enables remediation actions to be specified based on asset characteristics including:
  - Platform Type
    (e.g., Workstation, Server, Laptop)
  - Functional Category
    (e.g., web server, database server)
  - Organizational Boundaries
    (e.g., business unit, department)
  - Software Inventory
    (i.e., if a specific application is present)

Remediation Results

- Remediation Results convey the outcome (e.g., success/failure/error) of attempted remediation actions as reported by the remediation tool.
- 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

Open Vulnerability Remediation Language (OVRL)

- Provide the capability to express the low-level, machine-readable instructions necessary to perform a remediation
- 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

Additional Specifications

- CRE and ERI Data Exchange Formats
  - Basic XML representation that enables the exchange of CRE and ERI information within remediation workflows
- Remediation Tasking Language
  - Provides a standardized format to direct compliant tools to enact selected remediation actions on specific assets.

Comments

When developing these specifications we will take into consideration other open remediation languages, reporting or policy formats and tasking specifications being considered in the overall security automation architecture. This evaluation will include assessing conceptual alignment and the potential for schema reuse.

For more information...

Come to the Day Three Conference Workshop on Remediation. Some of the proposed specifications introduced in this presentation will be discussed in greater detail during this workshop series.

The remediation workshop track begins at 10:30am and concludes at 5:15pm on Wednesday, September 29th.

Contact Information

Chris Johnson
Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
christopher.johnson@nist.gov