Reinvigorating Remediation

Jim Ronayne, NSA
Kent Landfield, McAfee

SCAP Developer Days

Tuesday, July 10, 2012

Session Objectives

- *The remediation specifications have been slow to develop. The CRE specification has been released in draft form but has had little traction so far.*

- *This session will first briefly discuss a specific proposal for including CRE content in XCCDF benchmarks. A significant number of new benchmarks will be created this year and will include basic remediation content. A method for declaring remediation policy is required to make the content usable.*

- *The rest of the session will focus on the strategy for making remediation standards viable within the security automation community. Fundamental assumptions about future development and use of remediation standards will be questioned and a plan of action for specification, content, and tool development will be determined.*

First….

- How many remediation vendors do we have in attendance?
- Have you been following the Remediation Standards efforts?
- Does your products use XML or an ECMA-type language (proprietary or not)
- Did you participate in the remediation discussions?
- Have you read the CRE spec?
- Do you understand how it would be used and how it would impact your tool?

Using CRE

- Jim Ronayne, NSA

- *This session will first briefly discuss a specific proposal for including CRE content in XCCDF benchmarks. A significant number of new benchmarks will be created this year and will include basic remediation content. A method for declaring remediation policy is required to make the content usable.*

Using CRE

- CRE spec is draft but stable; needs implementation experience
- USG plans to create significant amount of SCAP content this FY; desire inclusion of CREs
- CREs will be created in DoD namespace
- Require a way to express remediation policy

CRE in XCCDF - Today

- Validate against 1.1.4 schema
- Include fixtext with a human-oriented description of how to apply the fix
- Use XCCDF variables (values) to set CRE parameters
  - Reuse check variable when possible, otherwise create new variables
- Use fix tag to indicate CRE and parameter
- Initial use case – human readable fix output lists
- Secondary use case – tools that consume XCCDF results and XCCDF fix policy

Example

<fixtext fixref=" cre_com.example_31-5_fix"> Set Domain Group Policy Object (Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options) using the IGroupPolicyObject interface. Network security: LAN Manager authentication level should be set to <sub idref=”lan_manager_authentication_level_var” />.</fixtext>

<fix id="cre_com.example_31-5_fix" system=”http://cre.mite.org/cre.xsd”>

cre:com.example:31-5:

lan_manager_authentication_level:<sub idref=”lan_manager_authentication_level_var” />

</fix>

CRE in XCCDF - Future

<fixtext fixref="maximum_password_age_fix">Set the maximum password age to <sub idref=”password_maximum_age_var_cre” /> days directly in the local SAM database (e.g., via NetUserModalsSet()).</fixtext>

<fix system="http://cre.mitre.org" id="maximum_password_age_fix">

<fix-export export-name="cre:org.mitre.cre.draft:var:117“ value-id="password_maximum_age_var_cre" />

<fix-content-ref name="cre:org.mitre.cre.draft:117" />

</fix>

Questions

- Should fix be defined in the XCCDF spec or separately (for use in XCCDF)?

- Should we start working on changes to XCCDF to support remediation now or wait until we have some experience with the current method?

Remediation Topics

- Kent Landfield, McAfee

- *The rest of the session will focus on the strategy for making remediation standards viable within the security automation community. Fundamental assumptions about future development and use of remediation standards will be questioned and a plan of action for specification, content, and tool development will be determined*

Derived Requirements for Remediation
(Draft NIST IR 7670)

- DR1. Method for uniquely identifying a remediation (CRE)
- DR2. Definition of an exchange format for basic remediation information (exCRE)
- DR3. Definition of additional data about a remediation, including mappings to applicable platforms, related vulnerabilities, or configuration issues (ERI)
- 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

Common Remediation Enumeration
(DR1 - Draft NIST IR 7670)

- Similar to a CVE

- 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 Data Exchange Format (exCRE)

(DR2 - Draft NIST IR 7670)

- An exchange format for CRE entries and related metadata 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.

Extended Remediation Information (ERI)
(DR3 - Draft NIST IR 7670)

- 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).

- 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.
- 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.

Extended Remediation Information Data Exchange Format (exERI) (DR4 - Draft NIST IR 7670)

- 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.

Remediation Policy Specification (RP)
(DR5 - Draft NIST IR 7670)

- 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 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.

Remediation Tasking Language (RTL)
(DR6 - Draft NIST IR 7670)

- 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 335 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.

Remediation Results (RR)
(DR7 - Draft NIST IR 7670)

- 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.

Open Vulnerability Remediation Language (OVRL)
(DR8 - Draft NIST IR 7670)

- 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.

So where are we?

- CRE format and usage described in NIST IR 7831.
- The CRE data exchange format is described Appendix B in NIST IR 7831.
- ERI will be fully described in a forthcoming specification.
- The ERI data exchange format will be fully described in a forthcoming specification document.
- The Remediation Policy Specification will be fully described in a forthcoming specification document.
- The Remediation Tasking Language will be fully described in a forthcoming specification document.
- Remediation Results will be fully described in a forthcoming specification document.
- OVRL will be fully described in a forthcoming specification document.

Questions

- How much automation do we expect to have?
  - Options might include:
    - notes for a user to implement,
    - output a usable file (e.g., GPOs to import, a kickstart file, a puppet file, a script to run),
    - output a list of known actions for an agent to implement,
    - output specific directions for an agent to follow.
- Which use cases do we want remediation to support? Which is most important? Which should we focus on first (i.e., which is most achievable in a short time frame)?
  - Configuration changes
  - Applying patches
  - Executing changes via a third party (GPO, WSUS)
  - Executing mitigating changes (locally or on a third party) where a subsequent rescan will not necessarily pass
- Do we expect tools to accept arbitrary remediation actions (within a prescribed set of possible actions) or are we okay with tools having to code to each new remediation? Or something in between? Are vendors willing to accept remediation instructions? How comfortable are vendors executing someone else's remediation actions that their staff have not had the opportunity to QA?
- ECMA scripting vs XML for OVRL?
- If XML, Would OVRL define fix actions instead of tests? A fix action would be implementing a given object and state. The object might not be the same as the check object (e.g., we might have a GPO object that gets implemented on a DC).

Questions (2)

- Is the existing CRE spec adequate to support building tools? If not, what pieces are critical to get started?
- Would you consider creating CRE content? For vendors - would you consider adding CREs to your content and associating them with CCEs?
- Do CRE parameters need to be more carefully specified? Is doing so essentially a start to OVRL?
- How precisely must a remediation action be specified? How precise must the method be specified in remediation (CREs intend to be very specific, OVRL might be less so like OVAL)?
- Who is responsible for:
  - issuing public CREs that are considered authoritative? CVE model will not work
  - Issuing public ERI information?
- Where is the boundary between existing remediation tools and the standards?
- Thoughts on integrating current SCAP-enabled compliance tools with the remediation standards?
- Of the component parts CRE, ERI, exCRE, exERI, RP, RTL, RR, which are the most important for us to focus on first? And why?

Questions (3)

- Strategy - the demand for remediation is growing and we need to start making actual progress. We've already done some proof of concept work. Can we agree on a goal for actual implementation (on a larger scale than what has already been done - and including some vendors)? If we make all this new content as described earlier will anyone build a tool that can do something with it?

- Do we need an incremental approach? Should we start by dealing with SCAP results and simply make it easier to generate a summary report that lists what needs to be on each box to become compliant (this might just require content with text fixes (which we have) and a stylesheet. It's not automation but at least the consumer doesn't have to go look up what to do. The next step might be outputing well forming instructions. Then output (for example) a GPO file that can be imported into a DC or a script that can be run. Again, these aren't full automation but they give the consumer a little more information and help them better use the results. Then we might focus on a local remediation tool (similar to the SPAWAR ref implementation). Next we would go for 3rd party remediation tools. Then we might focus on third party partial mitigations.

References

NIST IR-7831 – DRAFT Common Remediation Enumeration (CRE) Version 1.0

NIST IR-7670 – DRAFT Proposed Open Specifications for an Enterprise Remediation Automation Framework

Open Mic