Findings

Providing Actionable Data From XCCDF/OVAL Results

Findings

- Motivation – why is it needed
- Constraints on the architecture/design
- High Level Design
- Detailed Design
- Samples of Findings generated from XCCDF benchmarks
- Inter product operability (CFEs?)

Motivation – Why is it needed?

- Provide data to satisfy Auditors
  - Auditors often require more detailed information about a pass or a fail
  - A configuration item passes, but what value does it have?

- Provide data needed to remediate systems
  - Why does the antivirus check fail?
    - because there is no AV?
    - Is the AV present but not a valid version?
    - Are the signature files up to date?
  - Why does the file permissions check fail?
    - Do unexpected accounts have access?
    - Does the file exist?
    - Does each expected account have proper permissions

- Simply provide clarifying data about the state of systems

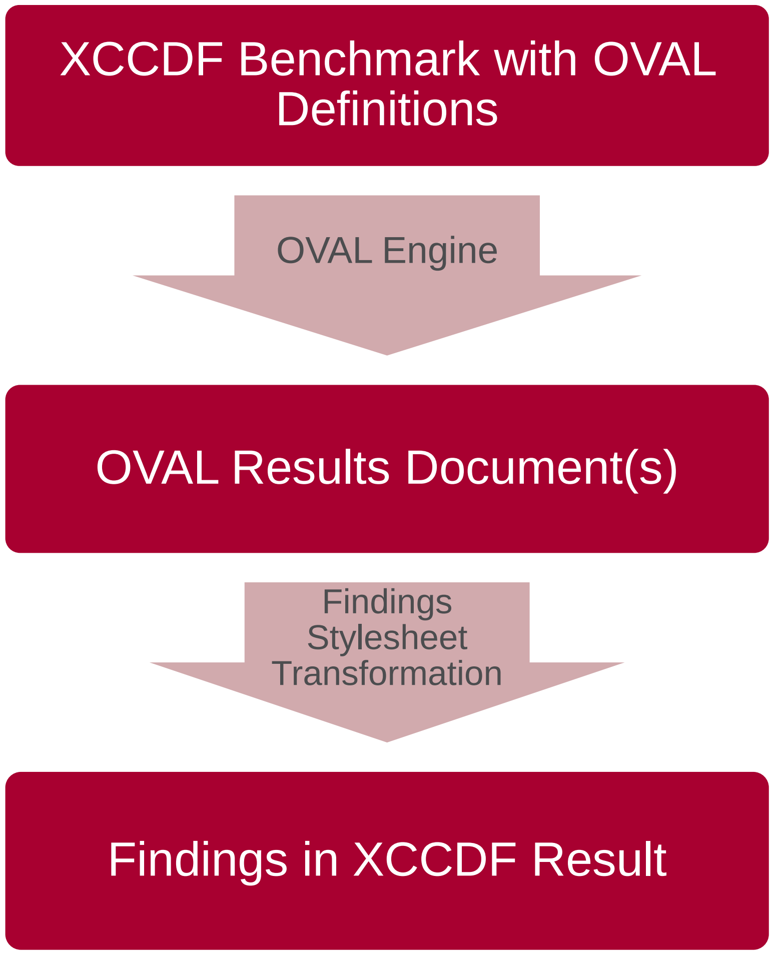
Constraints on the architecture/design

- For SCAP implementers
  - Findings must “fit in” with the rest of the SCAP infrastructure
  - Implementable with commonly available tools
- For Content Creators
  - Should have a low learning curve
- For SCAP Users
  - Should not require large resources at run time
  - Should reduce the volume of results to only significant data (high signal to noise ratio)
- For IT and Security personnel
  - Results should be clear, simple, and complete
  - Results should be localizable
- Appropriate for any checking system (OVAL, scripts, OCIL?)

High Level Design

- Process the OVAL results documents via XSL stylesheets to extract only the ‘useful’ information
- Each OVAL definition needing detailed results will have its own stylesheet
- XCCDF Results schema
  - We’re currently using the Check-content element as a container for findings
  - Rule-result should extended to provide a specific location for findings results

High Level Design


XCCDF Benchmark with OVAL Definitions
OVAL Engine
OVAL Results Document(s)
Findings Stylesheet Transformation
Findings in XCCDF Result

Detailed Design (1) - Components

- Findings schema
  - Supports instance data
    - Which file
    - Which account
  - Supports actual data
    - Actual permission collected for the file/account
  - Supports input (expected) data
    - The permission the file/account was expected to have
- Findings messages
  - Substitution for instance, actual and expected data
    - For file *xyz.abc,* account *USER1* had <u>read and execute</u> permission when **read only** was expected
- Mapping of OVAL Definition to XSL Stylesheet
  - Our implementation used an explicit mapping of ovalid to file name
    - oval:abc.xyz:def:101 to oval_abc_xyz_def_101.xsl
    - Xsl stylesheets are also stored in our database with the check id as the key
- Library of reusable stylesheets
  - Example - Many definitions check for file permissions, but a single library stylesheet template can handle all of them

Detailed Design (2) - Components

- For OVAL:
  - Use XSL Stylesheet to extract findings from OVAL results
  - Our implementation used an explicit mapping of ovalid to file name
    - oval:abc.xyz:def:101 to oval_abc_xyz_def_101.xsl
    - Xsl stylesheets are also stored in our database with the check id as the key
  - Library of reusable stylesheets
    - Example - Many definitions check for file permissions, but a single library stylesheet template can handle all of them

Detailed Design (3)

- Handle incomplete or partial results with attribute in the Findings document
- Indicate finding type (violation or compliance, and possibly others) with attribute in finding element
- Message and findings ids conventionally use URI style to provide for globally unique ids (not currently schema enforced)
- Finding messages are associated with a finding summary corresponding to the OVAL (or other) check id.

Findings generated from XCCDF benchmarks

- The account Power Users access to C:\WINDOWS\wmsetup.log is XRQNWATBDE(Modify) access, but no access is expected.
- The account Users access to C:\WINDOWS\wmsetup.log is XRQNE(Read&Execute) access, but XRQNWATBDE(Modify) is expected.

```
<findings xmlns="http://results.pa.mcafee.com/findings/5.2" id="oval:com.mcafee.oval:def:89558“>
        <finding isViolation="true“  messageId="com.mcafee.pa.msg.winfilenonerightsviolation">
              <instanceValue key="account">Power Users</instanceValue>
              <instanceValue key="filename">C:\WINDOWS\wmsetup.log</instanceValue>
              <actualValue key="permissions">XRQNWATBDE(Modify)</actualValue>
          </finding>
          <finding isViolation="true" messageId="com.mcafee.pa.msg.winfilerightsviolation">
              <instanceValue key="account">Users</instanceValue>
              <instanceValue key="filename">C:\WINDOWS\wmsetup.log</instanceValue>
              <instanceValue key="permissions">XRQNWATBDE(Modify)</instanceValue>
              <actualValue key="permissions">XRQNE(Read&amp;Execute / List Folder Contents)<actualValue>
          </finding>
          <findingsSummary isViolationSetComplete="1" totalViolations=“3"/>
   </findings>
```

Findings generated from XCCDF benchmarks

- The account Users access to C:\WINDOWS\help\ is XRQNE(Read & Execute / List Folder Contents) access, but RQNE(Read) is expected.

```
<findings xmlns=“http://results.pa.mcafee.com/findings/5.2 “ id="oval:com.mcafee.oval:def:89206“>
      <finding isViolation="true” messageId="com.mcafee.pa.msg.winfilenonerightsviolation">
          <instanceValue key="account">Power Users</instanceValue>
          <instanceValue key="filename">C:\WINDOWS\help\</instanceValue>
          <actualValue key="permissions">XRQNWATBDE(Modify)</actualValue>
          </finding>
</findings>
```

Findings generated from XCCDF benchmarks

- Password history length should be 6 or greater but is set to 0. (Failure)

```
<findings xmlns= "http://results.pa.mcafee.com/findings/5.2" id="oval:com.mcafee.oval.win:def:6001" >
    <finding isViolation="true" messageId="com.mcafee.pa.msg.winpasswdhistlengreaterthansetting">
       <instanceValue key="inputValue">6</instanceValue>
       <actualValue key="actualValue">0</actualValue>
    </finding>
    <findingsSummary isViolationSetComplete="1" totalViolations="1"/>
</findings>
```

Findings generated from XCCDF benchmarks

- Maximum password age should be less than 3888000 seconds (45 days) and is set to 3710851 seconds (43 days.) (Pass)

```
<findings xmlns="http://results.pa.mcafee.com/findings/5.2" id="oval:com.mcafee.oval.windows:def:17">
    <finding isViolation="false" messageId="com.mcafee.pa.msg.winmaxpasswdagelessthansetting">
        <instanceValue key="inputValue">3888000 seconds (45 days) </instanceValue>
        <actualValue key="actualValue">3710851 seconds (43 days) </actualValue>
    </finding>
    <findingsSummary isViolationSetComplete="1" totalViolations="0"/>
</findings>
```

Inter Product Operability

- As long as we have a consistent location for including findings in the XCCDF result doc, we will have a level of syntactic inter-operability

- However, to achieve semantic interoperability, we will need to have a common enumeration for findings.
  - Finding message ids
  - Standard substitutions

- Let’s look at the minimum password length as an example

Inter Product Operability - Example

- Vendor A might produce the following Findings document:

```
<findings xmlns= "http://results.findings.orgfindings/" id="oval:gov.usgcb.oval.win:def:6001" >
        <finding isViolation="true" messageId="com.vendorA.msg.invalidminimumpwlength">
          <instanceValue key=“required">8</instanceValue>
          <actualValue key="actual">4</actualValue>
        </finding>
        <findingsSummary isViolationSetComplete="1" totalViolations="1"/>
    </findings>
```

- Vendor B might produce this Findings document:

```
<findings xmlns= "http://results.findings.org/findings" id=" oval:gov.usgcb.oval.win:def:6001 " >
        <finding isViolation="true" messageId="com.vendorB.msg.minpwlenviolation">
          <instanceValue key=“expectedValue">8</instanceValue>
          <actualValue key="actualValue">4</actualValue>
        </finding>
        <findingsSummary isViolationSetComplete="1" totalViolations="1"/>
    </findings>
```

Inter Product Operability - Example

- Both results are valid and correct
  - For humans, they say the same thing.
  - For machines, they do not say the same thing.
- In particular, remediation engines would need multiple findings mappings to be able to remediate the issue
- For reporting there might be 2 (or more) sets of the same logical finding type
- To address these problems, we might consider CFE, Common Finding Enumeration

Inter Product Operability - Example

- A new enumeration – Common Finding enumeration

Status of Findings Today

- An integrated feature in the McAfee Policy Auditor 5.2 and 5.3 versions
- Being actively used by iPost today
- Extends the integration of OVAL and XCCDF to provide users with a missing capability
- Makes SCAP content more useful to customers without forcing them to munge XML results to get what they operationally need
- Being contributed to extend the SCAP set of standards
- Open specification is being provided not just to customers but to the community for others to integrate and benefit from

Questions ???

Kent Landfield – Kent_Landfield@mcafee.com

Dick Whitehurst – Richard_Whitehurst@mcafee.com

McAfee®