**March 20, 2019 SCAP Community Telecon**

**Security Content Automation Protocol (SCAP) Content Repositories**

The following is a collection of the key points brought forth in the two-hour telecon on March 20, 2019. The questions used to drive the conversation are highlighted in blue.

Charles Schmidt started the meeting by presenting slides showing how the SCAP content repository fits into the broader SCAP architecture. He explained how it would provide an interface to support both internal and external interactions that would hopefully provide a great degree of interoperability and automation. He talked about the different parts of interface I2 and explained how content gets from SCAP content providers to tools as well as those who are using SCAP content to create other SCAP content. He noted that a single protocol can handle both ends for publishers and consumers and that it would be a useful way to support this.

Who hosts SCAP content repositories?

DISA hosts a webserver of SCAP content files for DISA STIGs, which pushes content out to the community. It’s not very mature and they need it to be more automated for whatever toolset one is using but it is in use within the DoD. CIS hosts the OVAL community repository. Matt Martin pointed out that JP Morgan Chase has an RSA Archer based repository of SCAP Content. Siemens uses a Git-based SCAP content repository. Joval maintains a repository of both SCAP and OVAL content and provides a REST API.

Who uses a content repository as a consumer?

Joval uses content from well-known public OVAL/SCAP content repositories. They also support customers that use many of the well-known public repositories. Joval uses tools built by customers to pull content from repositories so that they can either leverage existing SCAP Content for reuse or update it. CIS uses a few SCAP Content Repositories.

Jessica Fitzgerald-McKay (NSA) mentioned that she is not a producer nor is she a consumer of an SCAP content repository, but her agency does consume SCAP content. They are at this event looking for situational awareness where content fits within the overall architecture of SCAP v2 so they can make their own plans accordingly.

Who are the stakeholders of your solution? Are there special requirements those stakeholders need? Alternately, would something simple (e.g., an RSS feed) be sufficient?

Joval’s users have requirements about the packaging of SCAP content, but those requirements don’t need to be handled by the repository. Their users often need to identify the subset of the available SCAP package that is applicable to their environment. This could be done locally after downloading, but this could also be done by having a search and filtering capability in the repository itself. To further explain SCAP content that may go to tooling, Joval indicated that SCAP content is provided in big packages, but it is not efficient so there is a need to filter and sort it. This could be done by the SCAP content repository. Otherwise, the tools that consume the SCAP content need to support this.

Danny Haynes (MITRE) stated that SCAP content repositories vary in implementation from a webserver of files to web interfaces that support more complex queries.

Jessica Fitzgerald-McKay indicated it would be a good idea if the group could decide if the SCAP content repository should filter based on how people have subscribed to the SCAP content repository or if it should just be an RSS feed.

RedHat hosts their SCAP content on GitHub and release it at a regular cadence. They provide tar files, zips, etc. From their support process, they deliver their SCAP content as part of their software packages that are supported and maintained. Should this be a tool expectation or in the specification? In RedHat, you can get SCAP content as an RPM package.

Callers indicated that CIS, Microsoft, and Oracle are doing something similar. Microsoft is doing it with Compliance Manager and Oracle with Solaris.

Charles Schmidt observed that in this case content is tied to relevant software that the content applies to. As a result, there is no further need for filtering. When software updates, any relevant changes to SCAP content will be included in the updated package. RedHat confirmed that this was their current process.

Charles Schmidt asked what happened if the SCAP content changed independently of a software update (e.g., to correct SCAP content). Gabe Alford (Red Hat) said as part of their release process, they build a complete XCCDF file or SCAP data stream file. A user can download and run it separately from the RPM itself. From a CVE perspective, they have a CVE stream where one can pull it down.

Would it be fair to say that a standardized Interface I2 could be used to distribute combinations of software and SCAP content?

Gabe Alford said yes but wondered if other messaging tools could be integrated into a Continuous Integration and Delivery pipeline (RSS, Slack, etc) to however subscribers are tied in and alert them that there’s new content. A more standardized protocol could be used. Danny Haynes asked if one would be a preference, but Gabe wasn’t sure.

David Ries (Joval) mentioned that the RedHat webpage with packages is terrific. It has sets of content organized by platform or by security advisory. As a consumer, the content is simpler and requires less filtering because it is one platform and is authoritative from a vendor. However, it’s different when pulling from a community SCAP content repository because one may only want SCAP content from certain authors. Or, maybe you’re pulling SCAP content from several SCAP content repositories and then prioritizing. It may be different for other platforms.

Gabe Alford added that if you have design where Microsoft, RedHat, macOS, etc., are in one SCAP content repository, how do you filter? How do you merge? Where do you put it? Should it be located in SVN or some other repository?

How difficult is filtering, searching, and combining? Is this something that an organization would do? Or would they contract that out?

David Ries said filtering is a specialized skill and that their customers expect the tools they buy/use to handle that for them. Populating the tools is an automated process managed by the tool/service vendors.

Vendors that compile content for their customers can find this challenging. For example, to get CVE SCAP content from Cisco, one would need to pull from two sources then prioritize because there is duplicate content. There is overlap.

Charles summarized the use cases:

1. Some vendors are combining SCAP content distribution packages for the software
2. There are vendors pulling SCAP content from multiple SCAP content repositories, combining SCAP content, and vetting it as a commercial service for their customers.

Is there a role for standardization?

Charles observed that vendors who ship SCAP content along with their already established mechanisms for sharing software package and updates likely don’t have a great need for standards for content delivery. Likewise, if there are commercial vendors whose service to their customers is collecting, compiling, and filtering content and then feeding the result to their own tools for customers, this scenario also doesn’t seem to require new standardization.

Gabe Alford said that standardization may not be necessary, but, requirements around SCAP content repositories would be very beneficial. Is it signed by a key, etc.? He suggested maybe a best practice guide for SCAP content repositories.

David Ries said that Joval does lots of indexing and filtering. It’s not a business they want to be in, but the customers are asking for it. There is inconsistency between SCAP content repositories. Some have to be screen scraped. It is often difficult to determine when a repository is updated, and there is often no machine-readable content to determine what content applies to what. There would be a benefit to standardization for this.

Charles Schmidt said in addition to indicating signing, applicability, etc., there might be benefit in a standardized way to declare what constitutes your authoritative SCAP content, what is new, and what has been modified. Ideally, update summaries would be provided to consumers before they download it.

David Ries said that at a minimum, each SCAP content repository should provide a standard manifest that describes packages, description, and what it applies to. When asked how the applicability of SCAP content would relate to this manifest format, he said there are many CVE SCAP content repositories that have packages that look like they are designed with specific applicability (RHEL 5, 6, 7, etc.). The manifest could list this information with URLs, some sort of standardized indication of the advisory each applies to, etc. This could provide consumers with a way to navigate the content.

Matt Martin said it sounds like an education issue. In his experience going through an Authority to Operate (ATO) check, as part of the Risk Management Framework (RMF) process, he documented where he would go for RedHat, Cisco, Microsoft, etc. Typically, at Northrop Grumman, they just did whatever the vendor told them: locate and acquire the content and document it. It may not be the best solution, but it was something. There wasn’t much comparison across different sources.

Charles Schmidt identified three different ways that content providers might categorize their content:

1. Here is our SCAP content. E.g., we have content for CVEs.
2. Here is the applicability manifest. E.g., we have SCAP content for Windows 10, RHEL 7, etc.
3. Here is provenance manifest. Where did it come from, some is authoritative, or some is different? You want to be able to compare manifest to determine if not identical checks, etc., to see what SCAP content is needed.

Charles Schmidt said the impression he was getting is that standardization for the actual distribution of content is not as critical as the standardization for getting out metadata about content (applicability, manifest, etc.)

For those who have content they are publishing out, do you have a method to share some of the metadata? Are there other metadata you can think of?

Several indicated they do have methods of sharing some of the metadata. Bull Munyan (CIS) said the metadata is something that’s already there and pieces of metadata aren’t standardized. It would be helpful to have that type of standardized information to then start searching for specific content that one would want to find.

Jack Vander Pol (SPAWAR) asked if there is a way to use metadata or manifests to advertise content someone wants to sell. He stated lots of repositories are out there, but we don’t know what’s in them. A vendor selling content could pride manifest of metadata to indicate what content they have available to sell. Potential consumers could view this to determine whether they want to buy the content.

Bill Munyan stated an SCAP content repository level set of metadata could be queried for the repository and could be useful.

Bill Munyan and Danny Haynes talked about the OVAL repository and what is going on there as far as metadata (e.g., OVAL repository extension, affected/platform., etc.). They indicated we may want to look at how other people are using this extension point in OVAL as a starting point for what types of information we might be interested in.

Charles Schmidt observed that, in some situations, it could be useful to separate content manifest and use content metadata directly to get the content. He observed that, when SCAP content is part of software distribution, (e.g., RedHat’s RPMs), that the metadata feed might be separate. Gabe Alford stated if he could go into the SCAP content repository, he would just want to download the manifest, check out what he needs, then pull it down from there. As such, it would make sense to keep the two linked. Charles indicated separation would not be required but we might want to ensure that it was possible.

Danny Haynes mentioned signatures and licenses under which content is distributed. David Ries commented that Joval had a manifest with title, description, OVAL family, use case, information about format (definition file, SCAP bundle), information about publisher, license, and URL to download. He said he would share it on the mailing list.

Any other topics for discussion?

David Ries was one of the contributors to the design and implementation of CIS OVAL repository. He noted community repositories have problems from the perspective of authoring new content. For example, one might want to trace content dependencies, or know if there are multiple users of a content component who might edit it later

If your repository is concerned about contributions from a lot of people, have a tool that when definition is created/modified, repository will tell you if it’s used by other definitions. If you’re an author, you may want the data connections visible. Standards that help clarify these relationships for content authors could be useful.

Charles Schmidt summarized David Ries comment by saying there is a need for certain types of metadata that authors might need from an SCAP content repository and the needs of those authors differ from what content consumers might need. Things such as dependencies and being able to track that. David listed examples, but they would be classified as metadata and would be very useful on how that data is represented and collected.

The meeting was finalized with Charles Schmidt mentioning the upcoming workshop April 30 – May 2 and said that we were hoping to come out of those meetings with subgroups that will commit to working on solutions.