December 13, 2018 Community Telecon

Interface I2 – SCAP Content Repository

Presented By – Stephen Banghart (NIST)

Stephen Banghart provided an overview of the SCAP v2 architecture and highlighted where I2 fits in the architecture. Stephen noted that the interface is going to be the same for the Posture Collection Server and Posture Evaluator to talk with the SCAP Content Repository as well as for an SCAP Content Repository to talk with other SCAP Content Repositories. Examples of external SCAP Content Repositories are the NVD (CVEs, SWID libraries) and vendors wanting to provide SCAP content to their customers.

# Question: Demographics of members on the call?

* Vendors
	+ Adam Montville (CIS), Bill Munyan (CIS), David Ries (jOVAL), John Field (Pivotal)
* Content Consumers
	+ Bill Munyan (CIS), Jarrett Lu (Oracle), Jessica Fitzgerald-McKay (NSA), Andreas Steffen (strongSwan), Hugh Smith (???)

# Question: Biggest challenges today with distributing and ingesting guidance and SCAP content?

* David Ries - Packaging of content. With current SCAP content, you might have an entire repository of definitions, but people want to just run a subset of the content and it’s challenging for end users or applications sitting between them to figure out how to slice-and-dice content for these needs. The second problem we have to deal with is for people that want to combine content from different repositories as part of the creation of one result set (e.g. some rules from a STIG, a few from an SCAP Security Guide, and some custom rules). Not sure if this in scope, but maybe the repository could let you select certain content and combine content from different repositories?
* Josh Lubbell - Lack of content that is valid with respect to 800-126 especially for less ubiquitous platforms. I also agree with David Ries’ examples.
* Adam Montville - I second everything David Ries said.
* Stephen Banghart - Adam Montville, in previous call, you talked about zero days. Do you want to talk about that?
* Adam Montville - I wasn’t so much talking about zero days in the sense of malware, but rather getting content out to customers with new products and when it is needed. But yes, being able to more quickly and efficiently distribute content so it gets to customers is important to us.

# Question: Key Features of this interface to facilitate the SCAP content ecosystem?

* Stephen Banghart - Is the ability for a repository to slice-and-dice content a key feature you need?
* David Ries - Not sure that one is required, but the ability to get subsets of content from a single repository or get subsets of content from multiple repositories is more important for our customers.
* David Kennel - XML has very low human readability that impacts sites that are doing local customization and tuning of XCCDF settings.
* Stephen Banghart - So, you would like the content to be both machine-readable and human-readable?
* Adam Montville - I think I would prefer that the interface as we specify it has the capability to have multiple bindings (i.e., could use over a message bus, over a REST interface, etc.). In different deployment scenarios, one binding might make sense over another.
* Charles Schmidt - I was one of the original authors of the TAXII 1.0 specification and we made it support multiple bindings and then for TAXII 2.0, the community said no we want one standardized binding because it is less complex.
* Adam Montville - I would like to see a plug-and-play ecosystem of security tools that can find what they need, maybe a bus-like XMPP-Grid solution, maybe we are using DNS records to find the correct records and avoid all these point-to-point configurations between security tools. It seems like the operations we want can be portable to different bindings. We can do it over a message broker or over something else. Maybe Bill Munyan has some other thoughts on this? Does this help at all Charles Schmidt?
* Charles Schmidt - Yes. Do others see the ability to have multiple bindings helpful especially in the acquisition phase? For the record, I am not for or against the idea, I just have that past experience.
* Adam Montville - Maybe we need to just narrow this down to a single scope and then do something different for another scenario?
* Stephen Banghart - We need a solution that can cover other scenarios. We don’t need to do that right away, but the ability to extend to something else would be useful.
* Adam Montville - About David Kennel’s comment about XCCDF’s readability, I agree. We need to figure out what parts need to be human-readable versus not human-readable and where. Then, just let the stuff that is automatable be automatable.
* David Ries - I can throw out two more requirements for consideration.
	+ Being able to support trust and provenance. This is important when pulling in content from multiple repositories (e.g., I want to combine Cisco content with CIS repository content, etc.) and then allowing people to understand the data they are using.
	+ Lots of questions about tailoring of content like XCCDF and maybe that should be a function of the repository?
* Stephen Banghart - Yeah, that is worth thinking about and keeping in mind as we work this problem. At the end of the day, tailoring the information you want to get is another form of querying.

Stephen Banghart gave an overview of what I2 really is and what an SCAP Content repository is. Specifically, Stephen provided concise definitions to be clearer (e.g. repository vs interface).

* SCAP Content Repository: Query/response repository that manages and stores SCAP Content. SCAP Content drives collections and evaluations. Examples include: CVE (NVD), SWID product libraries, security configuration checklists, etc.
* Interface I2: Abstract interface by which the SCAP Content Repository communicates with other components.

Stephen Banghart also provided a comparison of both the SCAP Content Repository (low security information, instructions that drive collection/evaluation, etc.) and the CMDB (high security information, results from running collection/evaluation instructions, etc.).

Then, Stephen Banghart explained how an internet-connected SCAP Content Repository can retrieve, duplicate, or receive guidance automatically rather than having to do it manually. Thus, reducing the overhead and time of content delivery.

# Question: Do you find value in the inclusion of SCAP content repositories in the SCAP v2 architecture? Is standardizing the interfaces to this component a valuable effort? If so, why?

* Josh Lubbell - I see value in standardizing I2.

# Question: As a vendor, is this something you are already doing?

* Adam Montville - We have CIS Workbench which is where we do all of our work and where content is generated into automated content. We also have CIS-CAT which can run the content. Even internally, we would like to be able to reach back and get that content. We have customers that also want this. Being able to get the latest content is something that would be important.

# Question: As an end user, do you see any challenges or obstacles with standing up an Internet-facing interface to automatically receive SCAP content? What are your main concerns for something like this?

* Adam Montville - We would be concerned about who has access to what and if tailoring is done upstream (e.g., start with a CIS benchmark and then tailor it), they would not want that pulled back and we also have to deal with all the transport security issues.
* David Ries - I am a vendor, but we assist end users in pulling in and aggregating content. One challenge I see, which is more of an adoption challenge, is a significant number of the repositories are very simple and a simple set of content (e.g., Red Hat has vulnerability content with just links). If I ran that type of repository, I would have a very difficult time justifying putting any effort into doing more than that. I am not sure from their perspective there would be a benefit to providing these additional capabilities.
* Stephen Banghart - We have a hard time dealing with just the content coming in and out and having a standardized interface would be helpful.
* Adam Montville - If there are any end users out there, I would be interested in hearing what they think about automatically receiving content.
* Jessica Fitzgerald-McKay - I see value in using this for sharing and acquiring content.
* Stephen Banghart - Are there any challenges that would be a hinderance to this?
* Jessica Fitzgerald-McKay - I think most have been talked about on this call, but there is a lack of content people want. If it was simpler to share content, there might be an increased adoption of content.
* Stephen Banghart - It is also interesting to think about these simple repositories with links to content. They are simple, but also lack a lot of functionality end users might want, but, don’t know is even possible to get.
* David Ries - I just wanted to agree with that and to say from some end users that are moving content to endpoints, one problem with the current end users is it is difficult to know when content is updated other than just retrieving all the content and sending it out. In the CIS repository, we can get monolithic bundles by class or platform, but there is also a git-based mechanism for pulling content down and some customers are using it because it is efficient, and you can see exactly what has changed. So, if you have a 60 MB file and only 20 KB has changed, you can pull that down. You might want to consider something like this.
* Stephen Banghart - At NVD, we expose data in monolithic files and people just pull these down on periodic intervals which puts stress on our servers. So, it would be helpful for us.

Stephen Banghart provided an overview of RFC 8322, the Resource-Oriented Lightweight Information Exchange (ROLIE), and how it could meet the requirements for I2 and the needs of the community. Stephen also discussed what ROLIE is and isn’t. Key points include:

* ROLIE is a standardized data format and protocol, based on the Atom Syndication Format, for storing, retrieving, receiving and sending security automation information
* ROLIE is not a database standard
* ROLIE is flexible (RESTful HTTP API, supports various levels of security)
* ROLIE is compatible with any file/information type through extensions

Question: Are there other implementations or solutions out there that also meet some part of I2? Any features in those that are useful or important?

* Stephen Banghart - Adam Montville, you had mentioned CIS had a repository, could you provide information about it with respect to this?
* Adam Montville - It’s not anything standardized, and we are working to get the API right. It’s a REST-based system, but, it’s not very complicated. Just a few simple operations, how you can access it, etc.
* Stephen Banghart - Do you need more functionality? Or, is this simple solution good enough?
* Adam Montville - We are looking to constantly discover things as we go. Right now, simple is good enough, but that doesn’t mean we won’t need them in the future. Benchmark authors submitting content to NVD might also have needs.
* David Ries - I can share one thing that might be useful or of interest with the CIS-sponsored community repository on GitHub. It’s not standard, but in choosing GitHub, it’s interesting because it’s like picking news articles and you want to get updates. Another way to look at it is like source code and you want to see all version changes, have others pull in your content, etc. There are lots of challenges and things companies do to deal with source code like this. So, the repository stores the content as files on GitHub and we have an API for accessing it. This is just one set of tools for working with a repository of source code. You can also include third-party repositories and see their changes, people can contribute, etc. We have gotten a lot of benefits from using git, but there has been lots of tradeoffs too.
* Adam Montville - In CIS Workbench, you can tailor the content to your needs and people want to be able to version content and I think I would support this content as code approach.
* Michele Cohen - I was just wondering if anyone was familiar with the Open Service Gateway Initiative (OSGI). It’s an open standards modular system and service platform for Java. It has some unique services that would provide lifecycle and source content bundling, etc. I am not saying we should duplicate this approach, but wouldn’t it be nice to have these things. It would be nice to have a more robust system for content services
* Stephen Banghart - I think these examples (git, OSGI, etc.) are great and we should see what we like and don’t like about them and what we can learn.

# Question: Who hosts SCAP content?

* Adam Montville (CIS), David Ries (jOVAL)
* Stephen Banghart - Would you be willing to talk about your solution?
* David Ries - Yes, we host our own generated content that we sometimes mix with other content and just provide URLs to our customers. The content is often hosted in use case specific ways.
* Stephen Banghart - Is simple enough? Or, do you feel more functionality is necessary?
* David Ries - Right now, simple is covering our contract for a repository, but, as a tool vendor we are getting more requests from customers to be able to aggregate content and provide it to them.
* Stephen Banghart - With regards to the whole slice-and-dice and tailoring issue, I think this is something we would like to provide without killing our servers. Not sure if there are end users and if they have issues with the current monolithic content system? Or, is the current system working for you?
* John Field - The use case for incremental change is important.
* Stephen Banghart - This is one thing that came up with git that was an important feature. It also helps the server when you only have to give out incremental pieces. I wanted to clarify too, Michele Cohen, you mentioned things in SCAP v2. We are working on this as a community and if we can improve ROLIE or find something else that works that is great. I think standardizing this is valuable especially as pressure increases to add capabilities. I think it also has to be easy to stand up and use.
* Michele Cohen - Yes, this resonates.

Question: This interface is about driving an SCAP content ecosystem. Are there any must haves or must not haves? Any thoughts on I2?

* Stephen Banghart - The ability to slice-and-dice, incremental changes, etc. does that sound like a reasonable ecosystem?
* Jarrett Lu - Seems reasonable to me.
* Adam Montville - I think this seems reasonable. I think one thing we are doing at CIS is all the tooling and a cooperative ecosystem of tools and we need standardized content as well. However, I think there are subtle differences, but, basically the same thing.
* Stephen Banghart - I agree. I think there are semantic differences, but, close.
* Adam Montville - I think it’s bigger than SCAP, but how can we get all these security tools to support an enterprise in a plug-and-play manner? How do we get from writing policy to getting it out and deployed in tools as soon as possible?
* Stephen Banghart - SCAP v2 is born out of the same desire. It’s just bigger in scope because the world is bigger with all the things that we want to connect.

# Question: Open floor for questions and comments?

* Charles Schmidt - A question for the community. Whenever, I work on a standard for a server, people always ask how you would discover it. For end users and organizations hosting repositories, how much does a standardized discovery mechanism matter? Right now, you have a clearly defined route, how important is it that people discover your repository with less information? Or, is this superfluous because you have ways that you already advertise these repositories? I will treat silence as we don’t care about automated discovery mechanisms.
* David Ries - I hear questions on where do we find content, but I don’t think you would write software around doing this. There are only a handful of repositories. It would be nice to have more webpages to list these sources.
* Adam Montville - Right now, it’s just so small that you don’t need some way to share lots of content like bit torrent, but, until the content is easy to create, I don’t think we are at that scale yet.
* Stephen Banghart - This reinforces the crawl before walk but make room to enhance in the future.
* Charles Schmidt - Being able to collect information piecemeal was raised, but, would a standardized interface, with a way to index and discover what types of content are present on an endpoint (a directory service), be useful? I guess my first question is anyone providing a directory service for the content you are providing? Or, is it just "here is the tarball of the content"?
* David Ries - The CIS repository does it in this fashion. Part of the GitHub repository is a documented set of python scripts and you can index based on different things and query the database. You could for example say show me all content for Adobe products on MacOS.
* Charles Schmidt - Do you have any idea of how widely used those capabilities are?
* David Ries - They are used internally to query and build all the posted packages. I also know of end users and software companies that try to do the same thing with the same level of granularity, trying to pull down by operating system and use case. I don’t know that the capabilities (search by author and others) are used other than to generate top contributors.
* Adam Montville - We don’t have the insight into who uses these capabilities although we could ask to find those things out if we really want to. But, there is no mechanism to do that right away.
* Charles Schmidt - Do any end users find these types of capabilities useful (i.e. what is in the package before downloading the repository)? Or, is just here is what is available, and I will download it?
* Jessica Fitzgerald-McKay - We don’t do any indexing like that.
* John Field - One thing that I see from my customers is people are trying to drive everything through automated pipelining and want to eliminate the step of pulling down a zip, extracting it, and finding the needed content. Something like git might be valuable here. Basically, get the latest version and do that without manual intervention is important to us.
* Brady Alleman - I wonder if there is value to solving a simpler problem of content distribution (not to exclude a more featureful protocol). A standardized file/directory structure representing a collection of SCAP content, served by existing web servers and their protections, could be more readily implemented and adopted. The software to build the standardized structure could be run separately from the server and the output uploaded. The YUM package repository format may be a good example of a similar approach.
* Stephen Banghart - This would be another good thing to look at.
* Charles Schmidt - So, I can understand, are you talking about a standardized categorization scheme that all SCAP content could be sifted into across all vendors so once you identify a repository, you can then know there is a category for vulnerability, applicability, etc. categorization of content?
* Brady Alleman - Possibly. YUM represents package metadata. Like git, I think it would offload lots of the querying capability to the client rather than the server.
* Michele Cohen - Have you heard of Mercurial? Mercurial vs git?
* Stephen Banghart - I have used it before. They are similar and provide more or less the same capabilities, but, definitely worth comparing to git to see what makes them good.