December 6, 2018 Community Telecon

SCAP Content

Danny Haynes began the teleconference by emphasizing that the material previously presented and presented in this call from NSA and NIST represents a proposal rather than a final design. As such, we are seeking feedback and other proposals to better meet community needs. The creation of an SCAP v2 will require an engaged community to create a solution that is usable by everyone.

Danny continued with identifying key objectives of SCAP. Specifically, these were: 1) shared meaning between SCAP producers and consumers, 2) support of security automation use cases, and 3) allowing common content that can be used across multiple tools. Danny emphasized that SCAP and its component standards need to be specified and managed by the community through an open, consensus-driven process.

Danny explained that SCAP v2 seeks to be explicit about what constitutes SCAP content. Specifically, SCAP content covers processing instructions (both collection instructions and evaluation instructions) and identifiers expressed in an SCAP data model, SCAP results covers generated data (both collection results and evaluation results) expressed in an SCAP data model, and SCAP information covers both SCAP content and SCAP results. He emphasized that SCAP results are not required to be created through the use of SCAP content, and can be the results of any process that have been formatted to use SCAP data models.

Danny explained that SCAP v1 had a vision for an ideal ecosystem. This ecosystem included a range of producers and consumers of SCAP content that provided sufficient value to encourage participation in this ecosystem. In addition, the ecosystem had a feedback loop to help ensure SCAP content improved in quality over time.

Danny noted that several challenges have prevented this vision from being fully realized. 1) SCAP content is complicated to make, requiring expertise in both the domain of assessment and in specific SCAP languages, 2) that often there was a lag time between changes to SCAP to address a new need and the ability for tools and content to take advantage of those changes to meet that new need (e.g., the historical gap between the need for a new OVAL schema and the time when that schema is developed, adopted, and then implemented in tools), 3) that managing SCAP v1 repositories was challenging, 4) that SCAP v1 often required vendors replace existing internal standards with SCAP standards, and 5) that there was often a communications gap between customers asking for SCAP content/support and vendors, with the former not having a clear understanding of who to contact in vendors and vendors often claiming that they were not hearing a sufficient demand signal.

David Ries (jOVAL) felt that, while writing SCAP content takes training, it is not really a significant challenge to the adoption of SCAP. He felt many other technologies, such as HTML, are harder to learn and still used ubiquitously. He noted that many organizations have been trained to write content, and many organizations are producing significant volumes of content. He added that he does see ways to make SCAP content easier to write and that it would be great to do this. However, he didn't feel that it was reasonable to point to the challenge of writing SCAP content as one of the main reasons SCAP has not been more widely adopted. David added that he also felt that the other items Danny listed, while certainly challenges that would be beneficial to address, were not the main challenges to SCAP adoption. Instead, he proposed that he felt a major issue was the lack of vendor-authoritative content from some key vendors. He gave the example of Microsoft, which does not produce SCAP content for its products. He felt that Microsoft any technical challenges to producing content could easily be overcome by a vendor with Microsoft's resources, and thus something other than technical issues was likely the cause. David noted that RedHat was producing SCAP content while Microsoft was not – he didn't feel he had the visibility into the organizations to explain why RedHat saw value in creating SCAP content but Microsoft did not. David also felt that the lack of content for different types of devices, such as printers and network devices, was another issue that was impacting adoption.

Josh Lubell (NIST) expressed that vendors were more likely to produce OVAL than XCCDF. Since XCCDF was more user-centric, it might be helpful to make that easier to write. He also noted that it can be very difficult to locate relevant OVAL definitions within repositories, and making it easier to find OVAL content would also be a help. Brent agreed, nothing that it was important to consider the content creation/management problem both from a front-end and from an end-user perspective.

Conrad made the observation that opinions on the SCAP languages can vary based on one's experiences. He noted that Windows vs. Unix, command line vs. GUI, front-end vs. back-end can all impact how one perceives the complexity of the SCAP languages.

Bob asked whether there was any direct support for SCAP from Apple. Danny and David Kennel responded that, at this time, Apple does not provide direct support for SCAP.

Joe Wolfkiel expressed that complexity often comes from the underlying complexity of the concepts SCAP needs to describe. He noted that currently SCAP tends to return binary (e.g., vulnerable/not-vulnerable) results, but in many cases a more nuanced response is what is needed. Specifically, he noted that it can be hard to trace the underlying cause of a result of "vulnerable". It can also be difficult to prioritize responses. He noted that there might be a case where a single patch addressed 15 vulnerabilities while another patch only addressed one. The first patch would likely be the most resource efficient, but the current way SCAP is written and used can make it hard to discover that. He also noted that when people run SCAP content with different profiles, it becomes very difficult to compare the results directly. Finally, he noted that it can be very difficult to give SCAP results to people with moderate technical depth and have them know what to do with it. He felt it was hard to "dumb down" SCAP in a way that made important distinctions and context visible. He noted another challenge was supersession of patches – one patch might subsume and remove an old patch, but old SCAP content that searched for the old patch would then trigger because that patch was no longer present. He felt that users needed to be able to have better visibility into optimal risk mitigation and system impact.

# Question: What value does SCAP content provide to the community?

* John Field (Pivotal) - We are interested in supporting customers with heterogeneous architectures. In monocultures, there is usually a product that can handle many security tasks, but this is much harder in more heterogenous systems. In heterogeneous/changing infrastructure SCAP serves as a lingua franca. In this case, content writing skills are transferrable across environments. He later added that SCAP allows one to write new content if a vendor is not doing so. As such, SCAP makes it easier to fill coverage gaps.
* David Kennel – The ability to customized configuration checks (XCCDF). He also agreed with John's comment regarding support for heterogeneous environments.
* David Ries – We work with vendors that do vulnerability and compliance scanning. Vendors are generally adding SCAP support to existing capabilities. They generally do this for a couple reasons. 1) Software companies want to do high-margin software programming work and minimize manual work. Writing content is a brutal manual process. SCAP lets you use existing content so they can focus on the high-margin things like software development. 2) Some vendors adopt because important customers have required it. (E.g., some customers want to sell to the US government, which requires it.)
* Jarett Lu – Our company initially had a script-based approach for assessment, but we switched to using SCAP. We wanted to take advantage of the ease of sharing content.
* Bill Munyan (CIS) - For us, there is a component that is mission-focused – we wanted to get our benchmark content in the hands of vendors who can certify it and add into their tools. Using SCAP was better than using a proprietary format to convey this content.

# Question: How should we prioritize work going forward?

Danny outlined some of the main issues from the previous teleconference: timeliness of endpoint status, lack of standardized interfaces, sharing content between applications, reduction in false positives, and concerns about supply chains. He asked if anyone had thoughts related to these or other concerns.

* David Kennel – The fact that major vendors (e.g., Apple and Microsoft) are not participating is a killer. Bob seconded this concern.
* David Ries – It depends on the scope of SCAP v2 changes. If SCAP v2 is about dealing with immediate problems, then better availability of quality content for key platforms and expansion of platform support would be good targets. He notes that this content does not necessarily need to be created by the vendors themselves and noted DTCC and Secpod both created a lot of content. If SCAP v2 is considering more sweeping, strategic changes, then simplifying the formats and coming up with new ways to express assertions would be great.

# Question: Are there issues around being aware of available content?

Dave Waltermire noted that people had mentioned many sources of SCAP content, some of which he had been previously unaware. He asked if the existence of this content was a surprise to others, and whether content availability was a major concern.

* David Ries - We are mostly private-sector oriented. In the commercial space, the end-users are isolated from this discussion by the vendor. The vendors gather the content. The users don't care about how the content was generated – they just want to get the compliance results. In that sense, the consumers might not even know they are using SCAP – they just trust the vendor. The vendors find available content to use because they pay people to search for it. It would be better to publicize, but the vendor group is much smaller than users. The content isn't well publicized, but the few people who need to track have the resources to do so.
* David Kennel - From a .gov perspective non-authoritative content may not be approved for use in evaluation.
* Josh Lubell - My experience is that there are a number of sources of vendor-specific OVAL content, but this content is hard to find using federated search engines.
* Dave Solin (jOVAL) - Commercial end users don't care about the format of the content – they just want tools that work and they want one throat to choke if it doesn't. A vendor might be using OVAL and SCAP and hiring people to write it, but the users don't care. The US Government might be different – they care about SCAP.
* Joe Wolfkiel – The labor involved in having people gather the content they need would be significant. We want to pay someone to gather the content for us on our behalf. This could be a tool selection criteria. We do require SCAP support because we want to write our own content, but the content the vendor provides should be basically invisible to us – we just need the content updated and then channeled into the scans. I don't expect end users to be SCAP advocates.
* Nick D – We are a security research firm. The asset identification standard in SCAP hasn't been updated in years. One reason I don't see wide adoption in SCAP is that it doesn't focus on a specific area – cybersecurity is so wide and you have many things. I could see OVAL focusing on hardware hardening and vuln identification. I agree with Joe – vendors have their own tools and solutions in the form of log files, configuration formats, etc. The vendor needs to spend time to convert that OS-level data into SCAP. The overhead is high in standardizing security content. Thus, instead of competing with vendors on content standardization, maybe we should standardize risk calculation. We can do that by standardizing asset representation so we are making apple-to-apple comparisons. There should be interest in identifying and standardizing the asset representation format. Oil/gas, IEEE, and others already have standards like that.
* Brady Alleman - The lack of a full and current reference implementation is a gap for content development. For OVAL specifically, OVALdi does not implement all OVAL schemas, and has not been updated to the latest OVAL releases. A reference implementation helps to resolve ambiguity in the specifications, and identify correct behavior when differences in behavior from vendor tools arise. What can be done to ensure that an SCAP reference implementation is available and maintained? Dave Waltermire responded that there is a robust test suite for OVAL tools that should be able to support content testing. Danny Haynes noted that MITRE is no longer able to maintain the OVALdi, but that if others in the community would be willing to pick that up, they should contact us.

Dave Waltermire, following on Nick D's comment, observed that NIST is proposing SWID tags as a way of standardizing identification and metadata for software. He noted that SCAP has traditionally focused on identifying vulnerabilities in installed software, but really there are three times one might wish to do this: 1) In SecDevOps when one wishes to incorporate a package it would be good to know if the referenced package is the latest version and whether it has vulnerabilities, 2) in software procurement, one might wish to know if software one plans to acquire and install has any vulnerabilities ahead of installation, and 3) the traditional SCAP area of scanning for vulnerabilities in installed software. Dave felt that it might be possible to support all three areas, but OVAL vulnerability definitions will only help in the third area.

David Kennel asked how SWID tags would interact with software installed from source code. Stephen Banghart responded that tag generation is tied to the software build process and if one is building an application oneself, the tag should be generated as part of that process. David Kennel responded that this implied near universal adoption of SWID tags, which he felt to be an uphill battle. Dave Waltermire responded that significant benefit could be achieved if the top 20 software vendors adopted SWID tags, and that he felt this was not a tall order to accomplish. Andreas Steffan added that he had a SWID tag generator that could create a SWID tag from any Linux RPM or DEB package, thus providing near universal SWID coverage for Linux environments. He noted that it would be better, though, if Linux vendors generated and signed SWID tags themselves. David Kennel responded that he was more concerned with software that was not handled by the major distros, for example, if someone installs a vulnerable application from a tarball. Dave Waltermire noted that a tarball could also contain a SWID tag file, as could software that was copied from a shared filesystem.

# Question: People have agreed that there is complexity in SCAP languages. What elements of it make it complicated?

* David Ries – We do a lot of training and have found the complexity lies in two places. 1) SCAP languages are highly relational. The simplest of checks require 6-9 objects that are all pointing to each other, and it gets much worse when trying to create more complex statements. It is basically trying to shoehorn a relational graph into a hierarchical format. 2) When you are creating real-world content, the languages support programmatic elements like variables, flags, and Boolean expressions. In many cases, if you were writing code, the test expression would be in the form of a for-loop. However, trying to express programmatic concepts like for loops in an assertion-based format is complicated. Meaning becomes contextual. Again, one is taking a simple idea but expressing it in a format that forces a complicated representation. Leland Steinke agreed that an iteration construct in OVAL would be useful. Such a construct could handle tests like "every home directory must be owned by its associated user."

Charles Schmidt asked to what degree is the complexity of SCAP content due to aspects of the language vs. the fact that, as Joe Wolfkiel noted, the SCAP is attempting to describe concepts that are inherently complex. David Ries felt that for most vulnerability content, the concept was actually quite simple – it is generally something of the form "if you are running Software X version a.b.c, then you are vulnerable." Many tests are like that. Having a language that could express conditions this simply would be great. He noted that some configuration checks do get more complicated. For example, tests that attempt to map between users and rights tend to be complex. Joe Wolfkiel agreed with David. His system at DISA had a simple list of application version and patches and that 80% of test coverage could be accomplished simply by consulting those two fields. He agreed that most vulnerabilities are as simple as that, although he did note that there are many situations that produce false positives (such as when one patch subsumes another). He felt that 20% of checks were more complicated. He noted that SCAP needs to handle both the simple and complicated scenarios.

Tim Harrison noted that doing assertions from data pulled from databases can be complex. He noted it got complex dealing with Windows user and group rights due to the many different object types and internal variables. He suggested that being able to cast data types would help.

Leland Steinke observed that different organizations interpret configuration requirements differently, and in a different way from the way SCAP authors typically do.

# Question: For those maintaining repositories of SCAP content, what are some challenges/best practices?

* Tim Harrison – Typically, data in SCAP repositories is not in a native SCAP format. As such, it must be converted to SCAP formats when it is exported. This creates additional hurdles and effort.
* David Ries - I've implemented 2 repositories that store SCAP native content, but as individual elements. I noted two challenges: 1) There are many challenges related to packaging. There are many ways one might divide up content. (Platform, use, etc.) In practice, if you have a vulnerability scan, you want to pull for specific platforms/applications. That becomes hard because there isn't a standard way of tagging. Correct searching relies on consistent metadata. In practice, this stuff needs to be sorted by OS and application. That is a challenge because there is not a predictable way to do that. 2) Because of the relational nature of this stuff, it can be hard when processing a contribution or making a change. If you change an object that is referenced by many others, it can be hard to know the complete implications. Do I need to make a new element or can I modify an existing element? This is hard if you are managing the content or if you are managing a contribution that changes content. It is hard to know if changes match the intent for dependencies. Also, SCAP content doesn't diff well – that makes change management harder.

Danny Haynes asked how many people ran their own repositories. David Ries, Bill Munyan (CIS), and Brady Alleman all indicated that they did so.

Josh Lubell stated, "I wrote a paper earlier this year that presents and demonstrates an idea for making life easier for SCAP content authors. The paper focuses on the SCAP source data stream data model, but I think the ideas could apply to SCAP component content as well. The paper (title: A New SCAP Information Model and Data Model for Content Authors) will be available for free from NIST once it's published by Springer (see https://www.springer.com/us/book/9783030045364). Hopefully this month. Please let me know if you'd like a copy once it's available." Also, "Here's a paper on the challenges of authoring an SCAP configuration checklist for a specific scenario (robotic discrete parts manufacturing). This one's already published and available thru NIST at https://www.nist.gov/publications/challenge-automating-security-configuration-checklists-manufacturing-environments."

David Waltermire polled the group asking whether there was value in standardizing interfaces to repositories. Both Chip and Jerome Athias expressed that there was benefit. David Solin said it depended on the purpose. If one is simply sucking up all the content, then it doesn't really matter how you got it. Apart from that, one needs to know how to slice and dice the content, and that is hard. David Ries suggested that a standardized interface could help with that. He noted that today, there are interfaces that allow you to download CVE definitions using a parameterized URL. However, if you want to do a sub-selection or combine sources, then this breaks down. He felt that if there was a basic API for sorting and filtering, that would solve some real problems customers have been experiencing. Right now those customers have to do their own filtering. Joe Wolfkiel noted that DISA just posts STIGs on a web page and you collect them using a URL. He felt it would be useful to have a scheme to notify vendors about new content or changes to existing content. For the latter, being able to point to what changed in the content would be very helpful. At the moment, there is a cottage industry of people creating homebrew solutions for this.

# Question: What make SCAP content reuse challenging? Is there any real value in content reuse?

* David Solin - RedHat has a rule where you cannot have any files in X. That is complicated. There is no way to pick that up and transplant it. When we run into problems like that in the context of generation, we just come up with a routine to express the complexity and then have a one-line way to generate that content. That solves reusability. Since end-users won't create content from scratch, I'm not really convinced they care about reuse. I know people want to tailor STIGs but I would not expect someone checking their system to rip open a multi-MB file to find content.
* Joe Wolfkiel – We actively discourage people from changing content since it complicates the understanding of the results.
* Jarret – We have customers who both tailor and who create their own content.
* Brady Alleman - OVAL makes difficult the ability to reuse the procedure of a check. You can reuse objects, states, etc., but these will often be what changes between related definitions. The ability to parameterize OVAL definitions (not OVAL variables as currently implemented, which can only be defined once, without external support) would be a considerable benefit to reuse within a piece of content.
* Tim Harrison - Even though there is some reuse ability in the language, people write to make very function-focused. People don't use XCCDF variables – instead they hard-code values in OVAL.
* David Waltermire – Software libraries that are intended for reuse have specific access points. The same is not true for OVAL. Also, some of these issues seem to be more about sharing than reuse. It might be good to have a conversation on that topic.

# Question: What are some issues around sharing SCAP results between tools?

* Joe Wolfkiel - We have a cottage industry of people taking data out of tools and using them in other contexts. We are trying to build systems that can take arbitrary XCCDF results. This is challenging given variables, different profile selection, etc. We want to believe that if we take this from multiple tools they mean the same things. The more the content changes the more important it is to capture content version/profile/etc. Also, native XCCDF/OVAL results are really big and it is hard to pick subsets if you have 200K endpoints. Tracking a failed rule to its ultimate cause is also hard. Danny Haynes noted that the issue of result size might be helped by event-driven reporting. Joe Wolfkiel noted that, currently, no SCAP tools support event-based reporting. He noted that some efforts have been made in that direction, but that filtering out specific changes is expensive.