IETF SACM WG Virtual Interim Meeting

May 2016

Notes as Submitted from Notetakers

Agenda bashing

WG Status

[Karen O'Donoghue]: Getting requirements to IESG is on my plate. There has been some discussion from IESG regarding use cases and requirements and whether they should be RFCs. My feeling is to go ahead and submit and see if there is push-back. There might be.

<ok>

[Adam Montville]: SWID call for adoption. During IETF 95 we identified some things moving forward. The only thing we have actually done is continue pressing on the IM. Did some work on Vuln Assess. We also decided how to organize drafts on GitHub. We still need to call for adoption on SWID M&A. In the room at IETF 95 seemed to think that would be good, but needs to be done. That is our ball – we dropped it.

Vulnerability Assessment Scenario Update

[Danny Haynes]: Status update – draft was adopted and we discussed at IETF 95. Document is now on GitHub so it can be worked there. Adam provided lots of feedback – good improvements. We responded with comments of our own – that is what we will discuss today. That discussion is on the list – look at them.

[Danny Haynes]: Managing terminology – holdover from back when the doc was originally being reviewed. We had some terminology in the doc and in the event it was adopted we planned to put that into a separate terminology doc. Running through these terms... Vulnerability description data = report. Vulnerability detection data – how to check for the vuln. Endpoint management capability = responsible for managing endpoint identity and providing basic information over time. Vulnerability management capability = take the vulnerability description/detection data and perform assessments. Vulnerability assessment = describes assessment process. Targeted collection = new one based on text in the document. Identifying some endpoint you care about and figuring out what information to collect and where to get it from. This is all just paraphrasing. Ask: do others think they will be used beyond this document. Maybe some could be used in use cases, although not sure we will update. Any thoughts?

[Dan Romascanu]: I'm concerned with some of these words because of the use of very general terminology. Need to be clear. In the terminology one we have definitions of endpoint and vulnerability. But use of vulnerability management is specific to this draft. They are used in a very specific way in this doc. Not sure they would be used in the same way without the context.

[Adam Montville]: Would you object to these terms eventually moving to the terminology draft. This draft is very narrowly scoped. My intent is to go after configuration assessment and management. I'm trying to bring to the table the concept of management capability in the context of a security program area. Eventually we will need this management capability defined in our context. Qualifies of vulnerability, configuration can be added.

[Jessica Fitzgerald-McKay]: I agree with Adam's vision. It is a capability that needs to grow, but there is lots of room to grow. Make sure in the future these terms are reused.

[Adam Montville]: So does it stay in the draft for now, or pull it out now. Another thing this will effect – capability discovery = learning what components in the ecosystem can do what. Might be relevant to this discussion.

[Danny Haynes]: Discovery is one thing that we are trying to find as part of the SACM task. There may be more on that at some point soon. Dan – does your opinion change.

[Dan Romascanu]: My instinct is to keep them in this draft. Leave here and say something like, in this draft the understanding of vulnerability management is ...

[Danny Haynes]: Seems reasonable. We can revisit when we evolve into other areas.

[Danny Haynes]: Clarifying vulnerability detection data – just defined as a representation of vulnerability description information. We were wondering, is this instruction that a security tool can take/ingest this and perform a vuln assessment based on it, or something else? If it is that, is it "guidance"

[Dan Romascanu]: You have an example in 4.2 – you speak about an XML-based language. Is this the only way it is implemented, or just one of?

[Danny Haynes]: From the perspective of an OVAL assessment – that is the only one submitted at this time. SWID M&A can collect software information. What else were you thinking?

[Dan Romascanu]: I don't have a complete code example. You are speaking of external vulnerability data. I'm trying to understand. How is the detection data combined with description?

[Dave Waltermire]: Some vendors publish machine readable vulnerability data. NIST has the NVD. We provide data feeds that describe vuln and characterize by the vulnerable products that relate to the vulnerability. Using that as an example, the tool would consume the NVD vulnerability records, compare that against their software load (collected by SWID M&A) and based on that comparison they would know if the vulnerable condition was present on the device. In some cases you need more info – is a feature turned on? For something like that you can use OVAL or some other collection against a device. That's how this example is intended to work.

[Dan Romascanu]: That helps. Thanks.

[Danny Haynes]: Based on that additional information, any other thoughts as to whether this vulnerability detection data is the content the tool consumes to drive its actions for assessment.

[Henk Birkholz]: I'm not sure. Guidance is steering the behavior of the consumer components. Maybe it is more like payload that is consumed – more like endpoint attributes. At the moment, it is more similar to the endpoint attributes that are collected rather than the guidance that steers the component.

[Dave Waltermire]: Does it matter if it is guidance or not.

[Henk Birkholz]: In the end it is handled differently. Might be useful to differentiate.

[Dave Waltermire]: One thing you touched on is an interesting way to describe – the vulnerability description data is metadata, but the piece that ties it to the vulnerable condition is essentially expected state.

[Henk Birkholz]: More like collected information about the current state. To determine if it is vulnerable you have to detect it.

[Dave Waltermire]: But the expectation is, if you have the state condition you are vulnerable. You are comparing collected state to expected state that is related to the vulnerable condition.

[Henk Birkholz]: That would help. Maybe generalize the question – expected state is guidance. In that case vulnerability detection data is guidance. Otherwise is not guidance.

[Ron Colvin]: The data is about a known-bad state. It is all information that is collected – you are comparing the known state to a vulnerable state and providing that input.

[Danny Haynes]: Sounds like you agree it is guidance.

[Ron Colvin]: Yeah. If you go looking for a vuln that is one thing. Other way is collect everything and comparing to a known vulnerability list. Could be asset information or vulnerability info. Every time you are saying it is a vulnerability you are putting guidance on what you have pulled from the devices.

[Henk Birkholz]: We can agree everything is prone to collection and attribution as vulnerability data is a flavor of this data. We don't only collect endpoint attributes, you also collect guidance – you need both to fuel a decision.

[Danny Haynes]: Defining targeted collection – More detail here. We had defined it as collecting specific information to make a determination about endpoint status. Question: You may be doing a vulnerability assessment and lack information so you do a targeted request. When we say targeted collection is it just a server getting information from endpoint, or does it have a broader meaning, or should we pick a different term. ("Supplemental collection"). Is "targeted collection" confusing.

[Henk Birkholz]: A collection is targeted – always directed at a specific targeted endpoint. I think is targeted is already in "collection". Later this assumption is violated.

[Danny Haynes]: Maybe hold off on this one and talk about it a bit more in the SACM task spot.

[Danny Haynes]: Processing vulnerability description information – The enterprise gets vulnerability description information from an outside source and puts it into a form its tools can use. Also assume the enterprise can extract endpoint information and make compatible with vulnerability description. Is this process of converting vulnerability description information into a usable format – Maybe already covered in previous discussions.

[Dan Romascanu]: Part of my question. Dave explained external information from a public vulnerability database. Implying that we are making this format converter – making this out of the mandatory flows in the enterprise engine.

[Danny Haynes]: We recognized it is needed. Out of scope for SACM.

[Dan Romascanu]: Assumption.

[Danny Haynes]: Yes.

[Dave Waltermire]: Document as an assumption.

[Dan Romascanu]: Yes. We have hit this again.

[Danny Haynes]: I'll bring these (remaining topics) up on this list. Some of the other ones are pretty straight forward.

[Danny Haynes]: Next steps – plan to do updates before the next virtual interim.

SWID M&A for PA-TNC Update

[Charles Schmidt]: SWID M&A was discussed at the last meeting by Dave Waltermire and I want to review the a few points of consensus from the previous meeting including a few half-open items that I want to talk about. Then we got a couple of new open issues that were raised since the preceding meeting. So to review, these are the items we have consensus. We are going to remove all references to TNC's IF-IMV and IF-IMC specification as it does not affect any normative capabilities in a problematic way. There was also consensus on retaining support for 2009 SWID tags and will fix up the language in the specification. Right now, it is indeed vague and we will make it far more concrete to say when you are using a 2009 spec, these are the fields you use and this is how you use it and when you are using the 2015 spec, this is how you use it. We will make that a lot clearer. Finally, there was a question about mandatory-to-implement (MTI) bindings when SWID tags themselves are conveyed. The observation was that today most SWID tags are in fact expressed in XML so it makes sense for XML to be an MTI binding. Our plan is to have a revised specification, with these changes, by the next virtual interim meeting.

[Charles Schmidt]: Moving on to what I call the semi-open issues where the consensus was either not complete or needs to be discussed. One of them was SWID tag versions. There was consensus that product versions need to be tracked and that is absolutely true, but, I think there was some confusion because SWID tag versions have nothing to do with product version tracking. The purpose of collecting

SWID tag versions is to facilitate metadata collection about software products and using endpoints as a source for that information. In other words, the idea of using SWID M&A to collect software inventory from the endpoints, SWID tag versions are completely unnecessary for this use case. So, I wanted to have a little discussion on this. The reason you would want the tag version is that the tag could contain additional metadata. For instance, some tags might include payload information with cryptographic hashes that are part of that software product. It is possible that a vendor might publish a product that includes a SWID tag and then at some later day, without changing the product, it might want to update the SWID tag that is associated with that product to include updated hashes, maybe there was an error one of the hashes, or they wanted to include an additional field. So, what they do is they push out an update and again this update doesn't touch the product itself, but, it changes the SWID tag to have new information. At the same time, it is possible that he server that is receiving and processing the SWID tag might want to use the endpoint as the authoritative source for the latest source of SWID tag metadata. They are waiting for the endpoint to get the latest SWID tag from the vendor and then the server asks the endpoint to see if it has a later version of the SWID tag. Then the server getting the SWID tag can tell there has been a change because it is a later version for that same SWID tag and can collect the SWID tag and use the latest metadata. That is the use case for SWID tag version. SWID tag versions have nothing to do with understanding which software products and which version of those software products are installed on the endpoint. So, given that, what are people's thoughts on SWID tag versions? Do we see the scenario described as something we want to support?

[Dan Romascanu]: My personal opinion is that we should collect them and document the delineation between the SWID tag version and the product version and make it clear those two versions are different.

[Charles Schmidt]: What is your goal? What are you trying to accomplish by collecting SWID tag versions?

[Dan Romascanu]: In the realm of an enterprise or multiple enterprises, you can have different versions right? Different enterprises using different versions. So a server interacting with endpoints, in a multi-site domain, you may have more than one. Is that true?

[Charles Schmidt]: So, yes, you could have multiple endpoints within an enterprise and the server is collecting the tags from all of them and it could potentially be that they are all running the same product and same version, but, it is possible that some might have received updated SWID tags and others might not have.

[Dan Romascanu]: Exactly.

[Charles Schmidt]: Why would you want to know that?

[Dave Waltermire]: The primary reason why you would do that is because the enterprise may be only getting metadata in the SWID tag, for the first time, through a collection mechanism. And if there isn't some way of detecting there is a new version of a SWID tag on a device that differs from the older version that you already gathered the metadata from, you need some way of detecting that so you can acquire that additional metadata. To me, that is the primary reason.

[Dan Romascanu]: Right and it also depends on why you would actually like to know this information. I don't believe the operator or the consumer needs to know, but, the software or the procedures that deal with the collection and integration of the data need to know.

[Charles Schmidt]: So, it sounds like there are at least two in favor of collecting the SWID tag version. Any other thoughts on that?

[Ira McDonald]: I am also in favor.

[Charles Schmidt]: That is sounding suspiciously like consensus. There will be some technical details that we need to work out, but, we will put together a proposal and run it by the list.

[Charles Schmidt]: The second semi-open one, I am not sure there is really that much to discuss. There was consensus that SWID M&A needs to support different bindings beyond XML such as CBOR which Henk proposed or maybe something else and that we want to clearly identify which binding is being used, but, we are going to have to put together a technical proposal on exactly how to get that information conveyed.

[Kathleen Moriarty]: To avoid confusion, can you switch your terminology and not use binding for data format because that would be more protocol things typically in the IETF to make sure if people come in from other groups and they are helping to review that the terminology stays consistent.

[Charles Schmidt]: Absolutely, which term should I use?

[Kathleen Moriarty]: Data format.

[Ira McDonald]: In S.A. hardware standards we are working on, we call it encoding.

[Kathleen Moriarty]: That makes sense too.

[Charles Schmidt]: The tricky thing is that I am already using encoding to say it needs to be UTF-8 characters.

[Ira McDonald]: In that case, call it data format and forget what I said.

[Kathleen Moriarty]: Data format would be more consistent across the IETF. Karen, that is right? I know you are in multiple groups that deal with this.

[Karen O'Donoghue]: Yes.

[Kathleen Moriarty]: Thank you.

[Charles Schmidt]: Thank you all. That is a new change I will have to make, but, it is not too much of a lift.

[Charles Schmidt]: Those were the only semi-open issues that I still had. Any comments on those? Ok, we will move forward with technical proposals on those and have them for you before the next virtual interim.

[Charles Schmidt]: So, we have two new proposals software location and source management. The first one is talking about software location. SWID tags themselves don't necessarily indicate where a software product is installed. They can have certain information in optional fields. Especially in the 2015 specification, they are unlikely to provide the absolute path location. So one question is do we want to add a field to the messages used in delivering SWID tags to try and capture where these applications are actually installed as associated with each of the delivered SWID tags? Certainly, there a couple major advantages to doing this. One is that it gives useful information for follow-up activities like patching; then it is useful to know where it is. The second is that it would pretty much eliminate double reporting which is likely to happen when you have multiple sources. So, if your SWID tag collection that is delivered by the endpoint is both pulled by a source that scrapes the filesystem looking for SWID files and takes those files and might also go to the package manager and auto-generate all the information. You might do this because each source might not have 100% coverage because the database may not necessarily drop a SWID file on the filesystem. However, if a package manager does drop a SWID tag, then you get both SWID tags and because reconciliation of that can be very challenging. Right now, the SWID M&A spec explicitly prohibits comparing those two tags and dropping one of them because there is not a reliable way to make sure that was always accurate so we thought it would be better to double report than to lose a tag incorrectly. So, if we do the software location, the advantage is we lose the double reporting problem because both of those tags will report to the same location and we can easily reconcile that. The disadvantage is that since SWID tags to not inherently reveal the location of the product there isn't always a mechanical way to understand where the product is associated with a particular SWID tag. I would be willing to say in most cases, you probably can, but, it is not always going to be possible. So in some cases, you may have inaccurate data or may just not know.

[Charles Schmidt]: So some questions for everyone, is software location important? Do we want to add it into the messages? And is the fact that is going to be about 98% reliable rather than 100% reliable going to be a problem?

[Adam Montville]: I think location is important at this point. A lot of the benchmark recommendations we make look for file checks and things like that require specific absolute path knowledge of where a specific file is located for some piece of software so it seems things like location would be important.

[Charles Schmidt]: Is a 98% vs. 100% accuracy a concern for you? It's an operational concern obviously, but, is it a concern from an engineering the standard perspective?

[Adam Montville]: I don't know to be honest. I think we would like to approach 100%, but, 98% still seems pretty good.

[Dave Waltermire]: We shouldn't let perfect be the enemy of good enough here. My instinct is that we are not going to get good location information in early implementations, but, that it is something that could be improved over time. If that were to occur, as we approach 100%, we get better and better information. The data that we collect becomes more and more usable.

[Dan Romascanu]: I am kind of wondering whether the percentage increases or decreases in time? Just to clarify, does this take into consideration virtualized applications or cases where the software runs one place, but, the data is accessed from a different location?

[Dave Waltermire]: The way SWID tags work, it wouldn't care where the data is accessed. It would basically just look at where the software is installed and run from.

[Adam Montville]: Right. Then, at that point, if the data locations are different and depending on the technology we are talking about, it probably has some configuration item in there saying where that data lives. So one example, in a MySQL deployment, there is a variable in one of the tables that tells you where the data lives and so that would give you the path. If you had the path to the configuration information, you would be able to get some of the other things you might need as part of a test.

[Jim Schaad]: When you say you don't have the location, there are two possible errors: (1) is you don't have the location and (2) you have the wrong location. Which one of those two are you talking about?

[Charles Schmidt]: I think in most cases it would be the former. There will be cases when you can collect the SWID tag, but, the SWID tag's location, the means by which it was collected, and the information in the SWID tag don't tell you anything useful where that is located. I don't think there will be very many situations. I am not going to eliminate them, but, I really don't think there are going to be situations where you get a location that is one place, but, it is actually in another.

[Jim Schaad]: I think that is much less of an issue then. You may not be able to fix a problem potentially, but, it doesn't tell you something wrong.

[Charles Schmidt]: Right, in that regard, we should probably have a means to report back that I don't have a clue where this thing is located so that the endpoint isn't just making something up to fill in the field.

[Charles Schmidt]: One other follow-on question is do we want to report the location every time a SWID tag is reported? Or, just when you are getting the full tag? You may recall SWID M&A can report tags as identifiers which map to full tags and is much smaller and network efficient and then there is the full SWID tags as XML. Should we make sure the SWID location always comes with the identifier as well or just include location if we are including the full SWID tag?

[Dave Waltermire]: One way of looking at this is if it is a two-step process where you get just the identifier and the location, then later you can request the full tag and what you would actually want to do is to send the identifier and instance with the request so that you are getting the tag associated with the instance.

[Charles Schmidt]: Right now, the way SWID M&A is set up is when you do what is called a targeted request where you want a specific tag, you identify the tag and not the instance and all instances of that tag get reported.

[Dave Waltermire]: This would allow you to get a specific instance. It could be optional and, if it is there, you could get the instance. Otherwise, you would get all instances.

[Charles Schmidt]: It sounds like at this point it would be worth exploring the technical details of how to do this because it would tell us the tradeoffs in terms of the control, bandwidth, and stuff like that.

[Dave Waltermire]: Yeah, we want to optimize so we are not always sending duplicate information when it is not necessary. I think that is the only concern.

[Charles Schmidt]: Agreed.

[Charles Schmidt]: It sounds like there is consensus around including support for software location to the extent that endpoints are able to associate it with specific SWID tags is a feature we want to add. Any objection to this this statement?

<no>

[Charles Schmidt]: Last issue is source management. I mentioned previously that SWID M&A can take SWID tags from multiple sources (package manager, file system, etc.) on a single endpoint. Right now, the way SWID M&A is designed all those sources get bundled into a single pile and that pile forms what is called the endpoint SWID tag collection and it is the endpoint SWID tag collection that gets monitored for changes and delivered during an inventory. So, there was a suggestion that we want to maybe start differentiating sources. For example, not just deliver a set of SWID tags, but, be like these came from the file system or these came from the package manager, etc. The advantage is if you care about the sources, as Dave mentioned earlier, they may be monitored at different rates. For example, a package manger could easily alert SWID M&A and say I just installed a package and here is all the information right away whereas scraping the filesystem may not occur in real time rather every 5 or 10 minutes.

[Dave Waltermire]: Those are good points Charles. There are also a couple other points of clarification. There are other sources of SWID tags on a system and I think this is really important if you consider the conversation we just had around location. In the case of a package manager or a filesystem, locations are typically going to be filesystem locations. But, there are other locations on a device where software can get installed. You can install software on an application server so there you need to look at the applications actually installed, you can install software in a database (e.g. stored procedures, etc.), there are also containers (e.g. Docker, etc.) where software can be installed on the system. So, the location you are going to report is going to be relative to where the software is actually installed. One way of looking at sources is as unique locations where software can be installed relative to the device and the location would be some path relative to that location on the device. Without documenting a source, you lose the ability to understand the location.

[Charles Schmidt]: So, in your mind, location is never sufficient to tell where a particular application is present?

[Dave Waltermire]: Well, unless we make it so. We would have to embed some sort of source context in the location in order to make it absolute relative to the device that you are talking about. So, one way or another, we will have to deal with the source issue. To your point earlier, the other advantage of doing source is how the changes to the SWID tag collection are detected also can matter from a source perspective. So some sources will only allow you to look at some previous state compared to the actual current state and compute a delta and report that delta. Other sources like a package database may record a transaction for every change that has occurred so you could report every increment of change that has happened from some previous point in time to the current time. Being able to differentiate what is actually being reported through collection and to be able to characterize what is reported along that dimension is useful because it will tell you how accurate the data is and the history of the device. I think there is a handful of reasons that we care about source.

[Charles Schmidt]: Any other comments?

[Dan Romascanu]: I am kind of getting back to my initial comment. We are not talking about software. We are talking about where the code resides, where it runs, and where it takes the data from. If we don't have all of this information, we cannot make a full assessment. Location is a composite of those things. In many cases, it can be just one, but, we need to be very well aware of what we are describing.

[Charles Schmidt]: SWID tags only do the first one. They are associated with software installed on an endpoint. They are not intended to have any correlation to is that code running or where that code gets its data. A SWID tag is only associated with the presence of a software executable on an endpoint.

[Dave Waltermire]: It does actually define where the code is located.

[Dan Romascanu]: Potentially source code location. We just need to describe this clearly that's fine. We just need to make sure we understand the concept that we are covering.

[Charles Schmidt]: Sure.

[Charles Schmidt]: I think we are out of time. So we may want to complete this conversation on the mailing list and if you have any additional thoughts, please share them.

[Charles Schmidt]: Just to wrap up, next steps, we are going to integrate the consensus changes I mentioned earlier, it also sounds like we have technical consensus on including SWID tag version and application location. We are still discussing source identification. We will make sure at the very least we have technical proposal for how you accomplish those three items by the next virtual interim meeting. Please keep the conversation going and any comments or suggestions are greatly welcomed.

Information Model Update

[Danny Haynes]: Status 1 – Many people have been getting together to merge I-D IM and WG IM. Currently working in Henk's GitHub repository. Also there were some concerns about IPFIX syntax. We are working on that.

[Danny Haynes]: Status 2,3 – Summary of work

[Danny Haynes]: Attribute and subject refresher – terms. One thing that should be considered as we work the merger – do these terms work? I'll make sure to put out on the list so we can get consensus there.

[Danny Haynes]: Statement and content element – Some newer constructs we would like to propose. SACM statement is a higher level construct containing metadata and content elements. Content elements are payload. Statement is shared between SACM components. Include which component collected, when, etc. Content elements have more specific information about what information elements were collected, when, where from, etc. The idea is that it can contain one or more content elements so you can bundle. Makes things a little more understandable.

<no comments>

[Danny Haynes]: I will send out an update of the WG IM with proposed changes so people can see and review.

[Danny Haynes]: SACM Relationships.... – Some new things. Relationships = construct to allow one information element to another. Two ways of doing: can embed unique labels in each information elements. Use as a key to relate. Other way is to provide enough information in the information element to do a content match. A bit more work, but possible. Events = express changes in Information Elements at a point in time. Requirements: must include new values/state of information elements. Optionally include previous values and when change occurred. Categories = construct to allow you to reference multiple information elements using a single name. E.g. "Timestamp" category.

[Dave Waltermire]: Timestamp information in second bullet for events – why is it MAY?

[Henk Birkholz]: The new value will have a timestamp. The MAY is for timestamps with past values. Timestamp of past values could be "last seen" or "created" – some details. New value has a timestamp.

[Kathleen Moriarty]: Going back to previous slide: This seems like this might overcomplicate things. We are just working on vulnerabilities right now. Do we want to do more on that before getting into this?

[Dave Waltermire]: This relates to the SWID M&A – SWID M&A is a way of communicating semantically equivalent information using a specialized data model. Maybe try to map this concept to SWID M&A and see where things fall out and report back.

[Kathleen Moriarty]: Wouldn't you have SWID inside a vulnerability anyway and not need a wrapper.

[Dave Waltermire]: There is the SWID information in a vulnerability report, but also the SWID information from the endpoint to some enterprise component. This content element really speaks to me as something that would be used to convey information from the endpoint.

[Kathleen Moriarty]: But is it necessary. May be some other schema. IODEF reference to SWID – pull that in directly. Not sold that this makes things simpler.

[Henk Birkholz]: We established some time ago that all information has metadata. It is always bundled. Maybe inside the content itself, but not all content can do that so it needs to be bundled with that content. Also, where data comes from (which component) is important and needs to be identified. Also, if there is conflicting information about an endpoint from two components, we need to explain how this conflict happened. So it isn't necessary for SWID/Vulnerability example, but you need this for some meaning of working information.

[Kathleen Moriarty]: I'm not convinced yet. We can see what the rest of the WG thinks.

[Danny Haynes]: Yes – needs to go to the list for more discussion

[Danny Haynes]: IPFIX syntax – All on this list so please chime in. We are looking at removing subTemplateList and subMultiTemplateList and creating our own constructs. Also adding cardinality of information elements – how many elements can be contained in another. Also other changes depending on the discussion. It's all on the list.

[Danny Haynes]: Next steps – Get open issues address by next VIM. I'll get a new draft out this week. Goal is to get the IM model stable by next IETF meeting in July. Let us focus more on data model development.

[Dan Romascanu]: Why is this intended to be an informational draft?

[Danny Haynes]: It just hasn't been updated. We talked about moving to standards track. Need to update the draft to reflect.

SACM Tasks Update

[Henk Birkholz]: While working on the IM there were some outdated tasks. These are current development question.

[Henk Birkholz]: Management of Target Endpoints over time – "Management of target endpoints over time". Have target endpoints that might be travelling, shut off, not observable in the domain. Trying to improve and give a bit more structure to collection and the complementary task of discovery. How to keep track of this?

[Henk Birkholz]: SACM tasks. Have this defined in terminology and use cases. The data task of second domain task is in the workgroup ID?

[Danny Haynes]: Not yet in.

[Henk Birkholz]: Tasks are the functions that reside on components that consumers produce.

[Henk Birkholz]: Goals – Goal of doing the assessment of target endpoints security posture. For that we need to know what a target endpoint is – where it is, what do you want to do in the assessment. Over time, the second task and keeping track of the endpoint.

[Henk Birkholz]: SACM Tasks and Target Endpoints – Different ways to get information from different endpoints. Different SACM components involved. The best way to think of a target endpoint is enterprise owned, maybe with a collector on it. On the other end of the spectrum – hardened unknown endpoint connected to the network.

[Henk Birkholz]: Given this range, there are Front line tasks – Targeted tasks: there is a targeted endpoint there is an initial knowledge.

[Henk Birkholz]: Discovery is pretty much untargeted. Everything that can provide endpoint attributes not about an endpoint you know but a new one. Then a more targeted task to get information, maybe only by secondary observation.

[Henk Birkholz]: Target endpoint needs to be identified and that is done by the Characterization Task. Doing the logistics on the back end. Provides a record of each endpoint encountered. Can add details if there is strange behavior or unwanted details. Uniquely identifying these is difficult – need unique attributes. Otherwise record might match multiple targets over time. Best effort process. In an ideal world everything might have a unique ID, but need to accept it might not be possible in all cases. In this case, a record might match more than one target endpoint.

[Henk Birkholz]: Examples – Known target endpoint with internal collector. Discovery might be initiated by the endpoint itself. There are many shades to this spectrum. Unknown would need to be externally discovered.

[Henk Birkholz]: All this populates Characterization Record. There were lots of discussion about how to label this. Include all identifying attributes in the record. But it is unrealistic to assume each target endpoint can be identified and recognized.

[Henk Birkholz]: Because target endpoints change and records change, there is an interesting case where you might think there are two targeted endpoints – 2 MAC addresses. But maybe later you realize this is one. Handle by merge based on knowledge from second domain. We are not prescribing any mechanics – just highlighting how target endpoints should be managed and taken into account.

<No questions>

[Adam Montville]: Let's review and discuss on the list.

WG Way Forward

[Adam Montville]: Want to keep pressing on IM. Danny has done a great job owning the draft. Would like to see a draft update by June 8 (1 week before next VIM). Need to get requirements to IESG – Karen is on top of that. The adoption call for SWID M&A is out. 1 week period on this – judge consensus. Also want to keep working on vulnerability assessment. Would love to get this into WGLC before IETF 96.

[Danny Haynes]: If we are targeting to get the changes in, open issues address. If we get those done by the VIM will that be enough time to get into WGLC.

[Adam Montville]: I think once the open issues are addressed we can put it into WGLC. I was thinking we could put into WGLC and discuss any major open issues at 96.

[Dave Waltermire]: Do you envision an ID update submitted by draft submission deadline and run WGLC from that point?

[Karen O'Donoghue]: Want a draft submission ahead of next VIM and based on that issue a WGLC. Would conclude in advance of 96. We don't want to run a WGLC a week before 96.

[Jessica Fitzgerald-McKay]: Reasonable deadline for input on the issues not discussed today?

[Karen O'Donoghue]: I believe during the discussion, the issues we didn't get to today will go out on email. If you go ahead and get that done, you have 3 weeks. Maybe get input on issues by the 31st.

[Danny Haynes]: I'll send out today.

[Jim Schaad]: Do we want to refresh the architecture draft?

[Adam Montville]: Personal opinion – expiration of architecture draft is inconsequential. Purpose was to have something to evaluate implementations against. We only have one implementation proposed. I don't see value completing.

[Jim Schaad]: That is fine.

[Henk Birkholz]: I think the architecture describes the whole components. Solutions mapped against the IM? Revising it now would only be necessary if we do something that violates it. We need consistency.

[Dan Romascanu]: Personally, I'm in favor of republishing with change. If it expires makes it harder to find. Should be out where it is visible by tools.

[Karen O'Donoghue]: My question based on Adam: if there is no intention to progress doc, does it need to be visible.

[Dan Romascanu]: I think so.

[Karen O'Donoghue]: Is there an intention to progress?

[Henk Birkholz]: Nancy had it on her agenda but hasn't had time recently.

[Karen O'Donoghue]: At this point we should leave it. If there is a need to revise we can do that. If there is no intention to revise I don't think it hurts for it to be out of sight.

[Jim Schaad]: I went to look for it, which is why I asked. If we think it will ever be published, I would like to see it refreshed.

[Dan Romascanu]: The way we discussed, we are postponing discussion. I think we need the information in the architecture as a reference. Separate from whether we publish as an RFC.

[Adam Montville]: So at this point I propose we complete this discussion on the list.

[Adam Montville]: One other thing on the Way Forward – Charles you mentioned there updating SWID M&A. Can you make that by June 8.