Developer Days 2013 – Minutes

OVAL Sections

Contents

Introduction

Developer Days 2013 was held from July 22nd to July 24th at MITRE's McLean, VA location. The event focused on OVAL, Remediation, and DISA's CMRS effort over the course of 3 days. The minutes provided here focus solely on the OVAL presentations, given by contributors across the community, including representatives from several commercial companies as well as the OVAL Moderators from MITRE.

The registration for the event exceeded 90 individuals, with anywhere from 65-75 people attending each day of the event.

These minutes attempt to capture the most important aspects from each of the 14 OVAL-related presentations given over the course of the 3 day event. In addition to these minutes, it is recommended that you also review the posted slides for each of the sections, in order to get a better understanding of the presentation.

OVAL Status and Section Objectives

Matt Hansbury
MITRE

Introduction

Matt Hansbury began the OVAL sections for Developer Days by first reviewing the status for the OVAL project covering the past year's events and noteworthy accomplishments. Additionally, he provided some high level objectives for the upcoming presentations on OVAL.

Details

Matt provided an update on the status of several portions of the OVAL project. OVAL is celebrating its 10th anniversary this year and in order to commemorate the occasion, the team had a set of coins printed to be distributed to significant contributors (both past and present).

Matt then discussed the status of the OVAL Language, highlighting milestones and achievements from the past year. He also discussed the next version of OVAL, 5.11, which is due later this year, with an updated draft coming in September.

Additionally, Matt talked about the OVAL Sandbox, specifically mentioning that the 5.11 Language development work has been ongoing within the Sandbox. While there have been many submissions to the Sandbox, there seems be challenges in getting things out of the Sandbox and into the official release. One of the day's presentations will discuss how to improve the Sandbox process to get new features into OVAL.

Next, short updates for the OVAL Interpreter, the OVAL Adoption program, and the OVAL Repository were provided, including significant milestones for each. During the past year, 3 minor versions of the OVAL Interpreter have been released. Also, the Interpreter passed a significant milestone in crossing the 30,000 downloads threshold.

Additionally, the OVAL team clarified the intended purpose of the OVAL Interpreter, both on its open source project page and also the MITRE hosted website. The team created and maintains the OVAL Interpreter as a reference implementation of OVAL. Some in the community have expressed concern that because the Interpreter is free for use, it can be viewed as a viable alternative to a commercial product. MITRE has never intended it to be a production ready or enterprise tool; it is primarily there to vet the language and provide reference for tool vendors and other implementers. The web pages and other documentation have been changed to emphasize that it is not intended to be used in a production environment.

With respect to the OVAL Adoption Program, several new companies have been added to the list of OVAL Adopters over the past year.

The OVAL Repository also passed a significant milestone this year, reaching the 15,000 Definition plateau. The MITRE team continues to process weekly submissions from the community.

Further, he provided a brief overview of the topics to be covered during Developer Days. There were numerous OVAL 5.11 proposed additions, several platform specific topics, mobile device topics, and a short discussion of how XCCDF, OVAL, and OCIL interact.

The OVAL Sandbox Process

Matt Hansbury
MITRE

Introduction

The OVAL Sandbox was originally introduced about a year prior with the goal of addressing several concerns. The previous method for making changes to the Language has been for the MITRE team to solicit input from the community and determine what the needs are for new tests, which would appear in the next schema release. This process required that the team have deep domain knowledge in many technical areas for an expanding number of platforms. This resulted in challenges creating and updating tests and made it difficult to have a scalable process. Additionally, it led to errors within the Language from time to time that had to be fixed in subsequent versions of the Language. The Sandbox addressed these concerns by being a location for anyone to submit experimental schemas to drive maturation of OVAL constructs, including existing elements as well as new requests. The goal of the Sandbox is to have these updates included in future official OVAL Language releases. This Sandbox has been used by multiple members of the OVAL community in addition to the MITRE Team. There are numerous examples of full schema updates and proof of concept code to implement the proposals.

Problem

There are two issues with the Sandbox process as it currently stands. The first is that while the quantity of content being put in to the Sandbox is commendable, the amount of feedback on said content is rather minimal. This feedback is necessary to evaluate and revise the proposals to where they could be moved forward in the Sandbox process. As a result, there has been nothing migrated from the Sandbox back into the Language. The lack of community review is causing a stagnation of proposals in the Sandbox, and effectively results in the same process as previous versions of OVAL.

Secondly, the rules for taking things from Sandbox and bundling them with a Language release are not defined enough. Without a concrete outline for how this is expected to happen, it is tough to objectively accept Sandbox contributions. Some things are universally accepted as beneficial to the Language, however there is also the possibility of moving something controversial into the Language by an ineffective process. A rigorous set of rules should be established to prevent this.

Proposal

To fix the first issue of lack of participation, the team proposed to use Developer Days and other OVAL-related public forums on a quarterly basis to generate consensus. The collaborative human presence factor is much greater motivation to drive progress than replies over a mailing list. The other option is to make use of the OVAL Board to help determine candidates to be migrated from the Sandbox into the Language. It was noted that one open question to this solution would be whether feedback is essentially limited to those present. The hope is to start the discussion at these events then continue the discussion on the mailing lists so that everyone has a chance to comment.

Discussion

Statement: The purpose of the OVAL Board is to make these types of decisions. Option one is not good enough except to drive discussion by the OVAL Board.

Response: Agree that the OVAL Board would be the clear driver for what gets put into the Language with option 2.

Statement: Do we really understand why there is light involvement? These types of things drive spikes of participation, but, is there a reason for light participation?

Response: To be honest, we don't know.

Statement: For those not aware, could you explain exactly what the OVAL Board is and how one would go about getting on the OVAL Board?

Response: The OVAL Board is a body of folks with deep domain knowledge, as well as in-depth knowledge of OVAL. They have shown a commitment to helping the Language grow and be successful. Determining Board membership has been a Moderator function up until this point, though the MITRE team is working with the Board now to empower the Board members to take over this role. There has always been a desire to have a good representation between government, primary source vendors, and end users.

Statement: The OVAL Board currently has members that have not been active in OVAL recently nor attend Board meetings.

Response: One other thing we are looking at is who among the Board is active and if it is determined that people are not participating, we will replace or remove them.

Statement: One reason we have a lack of participation is the lack of clarity on release timelines. With many priorities, if we knew there was a deadline for getting your input in on something, it would have a tangible effect.

Response: Agreed, just putting something in the Sandbox does not contribute towards a helpful review period. The OVAL team can provide more clear timelines around feedback to drive participation.

Statement: Two questions here. How do we ensure the contributions are high quality, and how do we determine what goes in the Language? From one person we hear about comments on quality, from someone else we hear about what gets included.

Response: These lines up with the two issues we are seeing.

Statement: How would OVALDI fit into this? It's one thing to say here's the specification, yet another to test it.

Response: One main factor that drives the Interpreter is to be able to test stuff like this. One thing we encourage for the Interpreter is to use it as a test bed, where you can implement it and test content. If people wish they can develop branches of the OVAL Interpreter to test with.

Statement: We have submitted an item into the Sandbox a few months ago, and the Sandbox is a sort of dumping ground. If they wanted to give feedback where would they give feedback?

Response: For one, the mailing lists are currently what we are thinking for soliciting feedback. It could be good to work with the author to identify the key aspects for what they want feedback on. We are having this talk to help get things out of the stagnant state.

Statement: Can such a thing be automated?

Response: Almost certainly yes. We get notifications when stuff is placed in the Sandbox and we can work on getting these notifications out to the rest of the community. We would like to encourage human interaction to send out the email notifications for announcements for feature sets or issues. People have been good about sending out notifications when they submit content to the Sandbox that points to the different aspects such as schema, sample content, and implementation.

Conclusion

While the section provided a few ideas for how to make better use of the great work that is found in the Sandbox, the community was mixed on how effective the suggestions would be in practice. The MITRE team will use the Developer Days sections as important feedback on features, but will continue to work with the larger community to ensure rough consensus before moving things into the Language. The team will also work on providing clear timelines around feedback expectations to help prioritize the review of new capabilities.

Lastly, the team will continue to consider how best to move forward with the overall Sandbox process to help fix the gaps identified, while keeping in mind the feedback from this session.

License and System-Metrics Tests

David Solin
jOVAL

Introduction

David Solin of jOVAL presented two proposed tests for the OVAL 5.11 Language release. Both of these tests have been posted in the OVAL Sandbox. The purpose of this section is to introduce the problem that the tests are intended to solve and show how the new tests solve their respective problems.

Problem

Each test is intended to solve a different problem, described below.

License Test

Windows operating systems store their license information in a large binary data structure available in the registry, found under HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/ProductOptions/ProductPolicy. The binary structure's breakdown has been reverse engineered and information about the structure is available online. The existing registry test can be used in order to make assertions about the binary content using complex regular expressions, but the resulting content is very difficult to write and to understand.

The contents of this binary construct can be very useful and in some cases required for testing things that David has found customers are asking for, including the determination of the licensing type of the Windows install, which determines certain features of the operating system software.

System Metrics Test

Windows operating systems also provide an API for accessing a series of system metrics including whether the endpoint is a Tablet PC and/or a MediaCenter PC, as well as many others. This information is only available via a documented Windows API (it does not appear in the registry).

As with the license information, David has found that a customer would like tests based on this information, which is not readily available via any existing OVAL tests.

Proposal

The proposal was to introduce two new OVAL tests to the Windows platform extension for the 5.11 release of the Language.

License Test

The license test includes the creation of an Object that would collect the license information and a State that would allow assertions about the various pieces of data that are found in the binary construct.

David showed before and after XML, showing how he and his team were able to execute the same checks in an awkward and challenging manner before the proposed test and then how the new test would encode the same information. The new test content was far easier to write and to understand. Additionally, David showed the group an extract containing the information encoded in the binary license information to illustrate the type of information made available.

System Metric Test

The system metric test includes the creation of an Object that would collect the system metric information and a State that allows assertions about the data found in the system metrics information retrieved via the API, referenced by name.

David showed before and after XML, showing how he and his team were able to execute the same checks using a small piece of custom helper code before the proposed test and then how the new test would encode the same test information, but with no helper code requirement. The helper application collected all of the system metrics and put them into the registry.

The new test clearly makes it easier and removed the reliance upon proprietary helper code.

Discussion

For the license test, some discussion around what types of information was available occurred. David made it clear that there was a great deal of information potentially available and that he was reacting to specific customers who have asked about evaluating some of the information made available by the test. He additionally suggested that one of the primary use cases was distinguishing between different licensing options.

How the existence of this information was determined was also asked. David suggested that the information, while not expressly made available by Microsoft, was available online. It was also pointed out that the group would need to do some research to determine what the values for this information should be.

Some tool vendors showed reluctance to implement this against an undocumented API. This was agreed to be a valid concern, though it was pointed out that this already does happen. It was suggested that interfacing with Microsoft to find out the correct way to check this information might be a good idea.

For both tests, it was asked if this information was available via PowerShell. For the license test, there is no known way to access this information with PowerShell and the system metrics test requires custom PowerShell. If there were canned scripts already available on the endpoint, that would be a valid implementation.

Conclusion

The general consensus in the room was that the tests were both of use and would be reasonable tests to add to the Language. The lone concern about proprietary APIs was conceded, but, the overall sense was that the positives outweighed the negatives.

Bitwise Functions

David Rothenberg
MITRE

Introduction

The need for a bitwise function in the OVAL Language was raised on the oval-developer-list when David Solin of jOVAL was trying to determine if the bitwise AND of two values was greater than 0 and discovered that there was not an easy way to accomplish this using the current version of the OVAL.

The original oval-developer-list post can be found at the following link.

http://making-security-measurable.1364806.n2.nabble.com/Bitwise-and-in-OVAL-tp7415374.html

Problem

In the current version of the OVAL Language, the “bitwise and” and “bitwise or” operations are available when working with integer data. These operations are useful when you are trying to determine if particular bits are set, or not set, in the collected value(s). For example, these operations will evaluate to “true” depending on if the specified value is equal to the result of the “bitwise and” or “bitwise or” of the specified value against the collected value. Otherwise, these operations will evaluate to “false”. Unfortunately, the way that these operations are designed, it is not possible to utilize the result of the “bitwise and” or “bitwise or” operations on the specified and collected value. As a result, it is not possible to solve the above use case using the “bitwise and” and “bitwise or” operations directly.

However, there is one approach that will allow you to solve the above use case, in the current version of the OVAL Language, and that is to carry out this check using the “bitwise and” or “bitwise or” operations and multiple states each of which represents a bit in the mask. The downside is that this approach is not very user friendly if you have a large mask with a lot of bits to check and requires that you know the mask prior to evaluating the OVAL Definition document. A simple example that demonstrates how to check the bits of a collected value against a specified mask, using the “bitwise and” and “bitwise or” operations and multiple states approach, was given.

Given the challenges associated with trying to solve the above use case, it was decided that the community should explore the development of a bitwise function that performs bitwise operators on integer and binary values and results in a collection of one or more values much like the arithmetic function that performs arithmetic operators on integer and float values.

Additional Background

A brief refresher on the different bitwise operations (AND, OR, XOR, NOT) was given. In short, AND will evaluate to "true" everywhere all the bits are set, OR will evaluate to "true" anywhere a bit is set, XOR will evaluate to "true" anywhere there is an odd number of bits set, and NOT will negate the input.

Proposal

As proposed, the bitwise function would consist of a bitwise_operation property that would specify which operation to perform (AND, OR, XOR, NOT) and it would take one or more arguments (i.e. OVAL Component or Function) as input. A simple example of how to use the bitwise function was given.

The operations available in the bitwise function fall into one of two groups: unary or binary operations. AND, OR, XOR are binary operations that can take 2 or more arguments as input whereas the NOT operation is unary and can only take a single argument as input. When working with the binary operations, in the case where there are multiple arguments with multiple values, the function will evaluate to a collection of values that represent the Cartesian product of the arguments. That is, each value in each argument will have the operation performed on it with each value in the other arguments. For example, if you have a bitwise function with the bitwise_operation="AND" and one argument with the values {A, B, C} and another argument with two values {D, E}, the function would evaluate to the collection of values {A&D, A&E, B&D, B&E, C&D, C&E}. Also, when working with binary operations, if both arguments have integer or binary values you will get results in the corresponding datatype. However, if there is a mismatch between the datatypes (i.e. one argument is integer values and another in binary values), the collection of values produced by the function should be of the binary type.

Next, the issue of how to handle truncation when using the NOT operation was discussed although it is primarily an implementation detail. The issue here is that depending on the limits of the internal representation of the integer or binary data on the system, you may get different values when you perform the NOT operation. Specifically, if you represent 0FFF as an unsigned short and negate it, you would expect the result to be F000 whereas if you represented 0FFF as an unsigned integer you would expect the result to be FFFFF000. Given this, it was suggested that everything extra should be ignored and the output should be truncated to be the minimum number of required bytes that you need to represent the input just so you don't get these additional bits at the front.

Another issue that was raised over the oval-developer-list and discussed is byte ordering. Specifically, it is not clear how values should be represented and interpreted. Typically, it is assumed the values will align with the target system, however, this can be challenging because the target system and the system that performs the analysis could be different meaning that values could be cast differently (little or big endian) resulting in different results. As a result, the OVAL Specification should be clear on how this situation should be handled.

Lastly, there is a reference implementation for the bitwise function using OVALDI 5.10.1.2 available on SourceForge.net (http://sourceforge.net/p/ovaldi/code/1678/tree/branches/bitwise_function/).

Discussion

Statement: Somewhat involved in the casting discussion on the oval-developer-list, I would suggest that if you are combining a binary and integer to just throw an error and rather than suggesting that you do what the platform does because everything isn't always running on the same platform. For example, we could be looking at an agent on an endpoint and evaluating it on a remote server.

Response: Yes, you could collect on one endpoint and analyze on another. This is completely up for discussion so I am glad that you are bringing it up.

Response: Exactly.

Statement: I think we should just say if you cast an integer to binary, in OVAL, you should use big endian.

Response: Okay, one opinion for throwing an error and another opinion for casting as big endian.

Response: As long as it's not based on the platform, I am fine.

Response: Yeah, I just want the specification to say when you cast from integer to binary, in OVAL, we use big endian.

Response: Okay.

Statement: If you make it platform dependent then you have unnecessarily complicated things.

Response: Very good point. We will consider it and discuss it with the OVAL Team since it is a big decision to say we are going to do this.

Question: It's not that big of a decision. Does everyone know what endian-ness means?

Response: David Solin had kind of gone over it on his last slide and you would have to do the reverse byte order…

Response: Yeah, right.

Response: So, I am glad you went first.

Statement: One potential extension for this proposal is the shift left and shift right operators where you can take a value and move the binary representation to the left or right a specified number of positions. I would imagine that we would pad with zeros on either end to avoid sign extension. Another extension that comes to mind now is if you want to add a big endian value to a little endian value, you might want to have a byte ordering swap perhaps that could be of use. I don't know if there is anything else that you guys are seeing.

Response: +1 on the shift because it allows you to extract a field and check it on a range.

Response: I just wasn't sure if this is something people would find useful.

Conclusion

Overall, there seemed to be a general consensus in favor of adding the bitwise function with the AND, OR, XOR, and NOT operators. However, we need to document how to handle casting between integer and binary values. There was also interest in further exploring the left and right shift operators as possible enhancements to the original proposal.

NT User Test

Jack Vanderpol
SPAWAR

Introduction

Jack Vander Pol presented an extension to OVAL that would allow reliable, consistent examination of Windows user configuration registry keys.

Problem

An automation gap exists in OVAL for checking Windows User Configuration settings. There is no reliable way to check registry keys in hive HKEY_CURRENT_USER (HKCU). When content is written against this hive, it is ambiguous with regards to what exactly should be checked.

Different tools return different results, because each is forced to make assumptions about the author's intent. Some tools ignore them entirely, some process the check for only the currently logged in user, while still others process it for all user profiles.

Proposal

This presentation proposes an OVAL schema extension intended to replace HKCU registry key tests. It proposes the ntuser_test that will read the ntuser.dat file for users who are not logged in and the registry for the currently logged in user. By creating an explicit test that has clear meaning and semantics, the ambiguity introduced by registry tests executed against HKCU is removed and consistent meaning can be achieved across tools.

Discussion

Question: Is there a case where accessing the NTUser.dat file will cause the system to lock?

Answer: Yes. You can't "load" it, but you can "read" it.

Question: The raw time format returned is in a unique Microsoft format. Should the time be transformed into a standard OVAL date-time format?

There was discussion of finding the user's last logged in time. It is useful to know if a user has not logged in in a long time and should/could be deleted.

Question: Can't this be implemented as a file test?

Answer: No. The currently logged in user info must be read from the registry; the ntuser.dat file is locked and unreadable.

There was extensive discussion on various complexities and issues with the implications of this test and implementation in a centralized management environment using roaming profiles. Consensus was that this use case is problematic and should be set aside for now, but, the test is valuable for individual box scanning.

Conclusion

Consensus from the attendees was that this test would be a valuable addition to OVAL.

Cisco Schema Updates

Omar Santos
Cisco Systems, Inc.

Introduction

Cisco is now providing IOS security advisories in OVAL and CVRF going back to March 2010 (http://tools.cisco.com/security/center/publicationListing.x). For the last three-and-a-half years, Cisco has been bundling security advisories on a semi-annual basis because of direct customer feedback as it makes things easier for large organizations that have lots of devices that they need to patch. Cisco is working to have advisories available the day a vulnerability is published. Furthermore, Cisco has been working with DISA to prototype the DISA STIG for Cisco devices.

In addition to taking a more active role in OVAL along with CVE and CVSS, Cisco is now using CWE internally and externally to educate the greater Cisco development community about common weaknesses that they are seeing as well as building lessons learned to teach organizations. Cisco is also working to adopt CPE and CVRF in IntelliShield which is a service that provides vulnerability information for both Cisco and non-Cisco devices.

Problem

In creating OVAL vulnerability definitions and working to prototype content for the DISA STIG on IOS, Cisco realized the existing schemas were limited and needed to be extended to support their use cases. The updates to the Cisco IOS schema and the introduction of support for Cisco ASA and Cisco IOS-XE are a result of this work and are proposed for inclusion in the OVAL 5.11 release.

Proposal

When Cisco began to take a more active role in OVAL, it was possible to check the version of Cisco IOS devices and check interfaces, however, it was very limited. There was also no way to assess sections of the configuration and check for things like quality of service (QoS), crypto, or anything else. The SNMP implementation was extremely limited as it didn't have support for routing protocols or to see if OSPF was enabled other than by running a command like "show ip ospf neighbor" or something like that. As a result, new capabilities were necessary.

Cisco also noticed that there were still references for CatOS and PixOS which are end-of-life and have not been supported by Cisco in quite a while now. Most Catalyst switches now run Cisco IOS or IOS-XE and PixOS was replaced by Cisco ASA. Given this, Cisco decided it would beneficial to enhance the Cisco

IOS schema and deprecate the CatOS and PixOS schemas. Cisco also added support for Cisco IOS-XE because some Catalyst switches (e.g. 4500) are now running Cisco IOS-XE. Cisco IOS-XE is a modular operating system whereas the classic Cisco IOS operating system runs as a service. Cisco is also exploring adding support for Cisco IOS-XR, unified communications, and Cisco NX-OS. However, support for these platforms will greatly depend on the community's adoption of the new Cisco schemas.

For the Cisco IOS schemas, the interface test was revamped and support was added for ACLs, routing protocols, and handling sections. Additional capabilities may need to be added since Cisco IOS keeps evolving, but, the current changes should cover most of what is needed. The changes are all in the OVAL Language Sandbox and have been in there for approximately 7 months now. Cisco is also working to further document the schemas once there is consensus to add them to the OVAL 5.11 release.

The Cisco ASA schemas support checking ACLs, interfaces, management frame protection, and SNMP. They also support the generic global, line, and version tests since they are not exactly the same as the existing tests in the PixOS schemas.

Next, an example of assessing a Cisco IOS device using jOVAL was given. This example involved running an OVAL Definition for CVE-2012-0381 on a vulnerable device and then running it again after the vulnerability was addressed.

Lastly, Cisco has been working hard to get a variety of resources out to the community including a blog, whitepapers, FAQs, and hosting webinars and is starting to receive feedback which is great. However, Cisco would like to encourage and facilitate more community adoption of the Cisco schemas. If anyone has ideas on how to do this or has any feedback or questions related to the Cisco schemas, they are encouraged to reach out to Omar Santos os@cisco.com.

Discussion

Question: Is there any place where you can download all of the OVAL Definitions at once?

Response: Right now, this is not possible. Every 4th week of March and September security advisories are published and are made available for download on Cisco's Event Response page. The URLs are also predictable so if you go to the advisory URL and append "/oval" you can just download the OVAL Definition from there. The challenging part is that it is not as simple as just updating HTML, but rather, involves a large ecosystem with a database backend. We are currently working to modify the tools and processes where you can download these files and are hoping to make a ZIP file of all of the OVAL Definitions back to 2010 available in a month or two.

Question: Are you using IODEF inside STIX/TAXII or are you using it outside?

Response: The goal, for me, is to carry an OVAL Definition in a CVRF document or at least point it to and at the same time carry it over IODEF and then over TAXII. Then, depending on implementation and who coded the tool, you may not have the two and that is some of the things that we are dealing with right now. The other thing we are working on is IODEF. Kathleen Moriarty, of EMC, is not here right now, but, is travelling to Berlin and is taking an active role in IODEF and chairs the MILE WG within IETF.

Panos Kampanakis, from Cisco, is taking an active role in extending IODEF to be able to carry anything. Does that make sense?

Response: I am working on the WG that is trying to extend IODEF and they are excluding some parts. Take a phishing email for instance. They have excluded parts of the phishing email from being part of the ability of IODEF to carry it and I have a little problem with that aspect of it. STIX will take IODEF in whatever format it is in internally; the problem is the community is all XML and you can do whatever you need to it, but, the community is pretty torn between the IODEF and STIX/TAXII thing. I think you are probably quite familiar with that. So, I was trying to get a feel for what path you intend to go down.

Response: I think it is too early to tell from our side. We can have an offline discussion on IODEF and STIX.

Question: If I am not mistaken, IODEF is not under SCAP?

Response: Correct.

Response: Neither is STIX or TAXII.

Question: Do you see SCAP trying to include IODEF as part of it?

Response: I cannot speak for SCAP, but, the security automation community is really looking into how we can operationalize these things and there are many others beyond this list. These are just the things that Cisco is getting into.

Question: For example, with routing, you are saying that the IOS schema applies to not just IOS, but, to these other platforms?

Response: I haven't got to IOS-XR and NX-OS because they are different. As far as IOS and IOS-XE, for routing protocols, think of how IOS-XE is running as a daemon on top of Linux. That is what it is right now and that is how we are able to modularize it and everything so the routing protocols syntax is exactly the same if you configure OSPF in IOS. It is going to be exactly the same as IOS-XE. NX-OS and IOS-XR are a different story. If we move forward and we create NX-OS and IOS-XR, which is what I was going to cover, we will see the configuration of the same things like router OSPF 10 or router EERP 25; it will be the same syntax, but, specific things let's say for BGP that are in IOS-XR and are not configured in the same way for IOS.

Question: There are other instances where a test from one schema actually applies to another operating system. They originally tried to lump these tests into the Independent or UNIX schemas. For instance, you might get the Linux RPM test and you can run that on AIX too because over time they added RPM to AIX. Is this cross-platform nature documented in the schemas?

Response: Yes. Once we adopt it, we will create better documentation with an example definition. I do have some documentation somewhat ready that we would like to share as part of the review. To be honest, there is a lot more return on investment if I can say I created this schema and organizations are actually adopting it so that is one of the things that we want to foster because we are spending a lot of effort and money building just the IOS definitions and schema. To give some perspective, we have to modify internal tools to be able to parse the several hundred lines of version information into some automated way that can convert it into XML and then have someone in the lab check to see if works or not. We have working with the jOVAL folks, and thank you guys again for all the support you have been doing, because jOVAL is the number one tool we are using for testing this content. Now, having said that, going back to IOS-XR and NX-OS and unified communications, we don't have support for that, but, we have had some customers already ask us for that are not in the government sector which is a good thing because we want the whole greater community to look into this. The problem is adoption so we want to see people at least using IOS, and eventually ASA, before we can actually move further.

Question: You suggested deprecating CatOS and PixOS. I know you said that they are no longer supported products, but are the schemas wrong for those products?

Response: The schemas were very limited. Wrong is a very strong word. In some cases, there were things not supported in the schema and it made a lot more sense to dedicate effort and prioritize the enhancement of IOS so we can then support the Catalyst switches 6500 and everything that is going to run IOS. There are organizations that are still using PixOS, and I am talking very old version of PixOS (e.g. 6.3), and it runs so they don't replace it. As far as vulnerabilities, I cannot tell you if they have it patched or not. For those, you can do some checks like the version check because it is very simplistic, but, it also makes more sense to dedicate resources and instead of updating the PixOS to just concentrate on ASA and creating the schema for that because the greater community is actually using ASA now.

Response: I was just thinking we normally just deprecate something with a warning that in a future release it will be removed from the language, but if it is still valid although it may be ancient use case, I don't think we would want to remove it. But, I kind of agree that we do want to say something about it especially if there is something that is not right with it.

Question: For 5.11, does it only have to be deprecated? Can't you only remove it in a major version?

Response: So for 6.0.

Response: Well, true. Just marking it deprecated now suggests it would be removed. And it's like, should it be removed? If it's not hurting anything then, but, if it is hurting something then yes.

Response: I will say, I don't think anyone is creating content for it anyways.

Question: If there is no interpreter, does it really matter?

Question: Maybe we need an abandoned keyword?

Question: So are you working with DISA on STIG content? I know you mentioned the STIG versus checking vulnerabilities?

Response: One main thing that sparked the interest of changing the IOS schema is we started working with the prototype STIG, with DISA, and we even identified some STIG checks that could be enhanced as well. But, even as easy as checking if the configuration has a specific state, it is not as easy as you may think because you need to use internal and external variables; it is a monster. You can check a banner very quickly, but, checking that a routing protocol is a more complicated thing. With MPLS configuration, you have several different ways that you can configure it. For IKE and IPsec, there are many ways you can configure an interface not even with crypto map. It is very dependent on the version if crypto maps or not crypto maps apply to an interface. GRE over IPsec, L2TP over IPsec, remote access VPN, LAN-to-LAN tunnels, there are plenty of ways Cisco developers have actually found a way to create new commands to be able to do this. Having said that, this is one of the reasons that we had to add that support today and we stopped creating the definitions until we had support in the language as well as community adoption. It is one thing to create the definitions, but I need to test them, and we need tools in the community to adopt them. The second thing is there are not really any authoring tools out there which would help operationalize this. These are the challenges that we are facing. Does that make sense?

Question: What interest are you seeing in adoption as far as NX-OS and some of the use cases your customers are bringing to you?

Response: To be honest with you, the NX-OS support has been the least of the platforms and I was expecting it to be one of the bigger ones. Data centers are extremely hot right now. Virtualization and SDN are going out of proportion right now from a keyword and buzzword perspective to the actual implementation of it. Switches and UCS are selling extremely well right now. I was expecting to get a lot of questions about if we are going to support NX-OS or not. As a matter of fact, I actually got more questions, especially from service providers of course, about IOS-XR rather than NX-OS. So, I haven't seen that much, but, we put it there because it just makes sense. We are investing in creating something that is very similar to IOS so we might as well create this schema especially since it takes so long. I created a schema and it's going to take iterations of adoption to have it in there. The other one that I have gotten feedback on is unified communications. That I can tell you will be tricky. The reason for that is let's forget about Cisco creating definitions and forget about different IP phones. Think of unified communications as a whole, you don't have one device or one product. You have IP phones, something to manage it and control the communication (call managers are what we call it), voicemail, and integration with multimedia applications (tele-presence and all that stuff) so it is a monster. So, even if you a vendor out there, it is probably one of the most challenging things to actually interrogate. Now, the good thing is there are a lot of the checks for operating systems and if there is already a schema for how we can interrogate the actual devices, that the community is already using, it would be great to not have to reinvent the wheel.

Question: Can you expand a little bit more on the STIG prototype content and how much coverage did you see in addressing this?

Response: We actually only did the infrastructure layer 3 router, we didn't do layer 2 yet. Out of the hundred-and-three or hundred-and-five STIGs, we did about eighty-one of them. We couldn't do the other ones. Some of which are completely impossible to do and we probably need to change the STIGs. The other ones we didn't' support in the schema. So, we completed the prototype and it exceed the expectations of what we were trying to do because when we looked at it there were a lot of hacks that we had to make with internal and external variables especially with limited support. It was a challenging process especially when we didn't have support for certain things so we decided to investigate the issues so we could enhance the schema. Again, I am an engineer and I cannot control how IOS is going to communicate over BGP tomorrow, or there may be a new extension of BGP that we don't know about, or there may be some new way to configure families for MPLS that gets implemented tomorrow and becomes a best practice, but, at least to attack it from a visionary perspective, we can interrogate the device and even if it is Cisco or non-Cisco, the community can create some content for it. I think it provides a little more value rather than try to hack one of the STIGs and that is why we prioritized that and we stopped the creation of them.

Question: Are there any plans to move forward and publicly release the DISA STIG content?

Response: We gave the content to DISA and they are still validating it, but, it is their content. If they want to publish it, we are okay with that.

Statement: Just to make one clarification about the deprecation policy. You can deprecate something in a major or minor revision, but, it has to stay in the language (in its deprecated state) for at least one minor revision before it is removed. However, in practice, we have only removed 1 or 2 tests out of the many that we have deprecated. So, it is not very likely that it will be removed right away. Chances are Version 6.0 removal probably makes the most sense.

Response: Once again, one of the major comments was does it hurt anything? It does not truly hurt anything especially if there is no content support for it, but, it is just a matter of if there is something that affects the critical infrastructure that is working with CatOS people can create their own OVAL Definitions. Let's say we were not creating OVAL Definitions for vulnerabilities, but, say you wanted to check something, anybody can create an OVAL Definition. As long as there is a schema and reference implementation, the sky is the limit. Whether it works is a different story. So, it doesn't hurt that much. As far as the cleanup process, that is something beyond me. Whether it makes sense to keep it forever and ever and to keep everything we deprecate, that's something else.

Conclusion

Overall, there seemed to be a general consensus in favor of adding the new features to the Cisco IOS schemas as well as adding support for the new Cisco ASA and Cisco IOS-XE schemas.

Checking Packages on Mac OS X

Jasen Jacobsen
MITRE

Introduction

Mac OS X uses packages to keep track of installed software similar to other UNIX operating systems. This presentation proposed an addition to the OVAL Language to access the Mac OS X package receipt database.

Problem

Enterprises wish to track the packages installed on their machines. Package tracking enables use cases such as scanning for known vulnerable versions of software installed across the enterprise or ensuring that only authorized software packages are installed. OVAL already supports similar functionality with other operating systems, for example using rpminfo_test and dpkginfo_test. It should be possible to use OVAL to evaluate information about packages installed on Mac OS X machines.

Additional Background

Mac OS X applications often use the Installer application to install themselves. Installer writes receipt information into a database. This information includes a ".bom" bill of materials containing a list of all the files installed by the package, and a ".plist" property list consisting of metadata about the package including the version and install date.

Proposal

Jasen discussed potential approaches to access the package receipt database. The OVAL plist510_test could potentially be used but is not recommended. Apple's documentation states to use the pkgutil command line tool to access this information, not to directly query the receipt files, as the location of the receipt files may change. Jasen discussed the package information available through the pkgutil tool.

Jasen proposed adding a pkginfo_test to OVAL modeled around the information made available through "pkgutil" and based on prior experience with the OVAL rpminfo_test for Red Hat Linux packages.

Jasen discussed that unfortunately, the pkginfo_test may not reflect the current system state, as the user may have manually deleted the application. However, the group was reminded that the rpminfo_test is widely used despite similar state issues. The Apple System Profiler is a potential alternative to using the package receipt database as reported by pkgutil.

pkgutil can regular expressions to search for packages, however, its regex parser may be different than OVAL's regex parser. Jasen suggested the OVAL interpreter mediate the regular expression parsing by fetching all packages from pkgutil and performing its own regular expression parsing rather than relying on the parsing done by pkgutil.

Discussion

Josh Wisenbaker and Shawn Geddis from Apple provided feedback:

- In some cases, even if the application's uninstaller was run, the package may still show up in the pkgutil database. pkgutil provides historical data about what has been installed. System Profiler shows real-time data about what is installed.

- Not all applications use Apple's Installer. Applications that do not use Apple Installer will not show up in the pkgutil package receipt database. Instead, System Profiler should be used to determine the installed applications, while pkgutil is useful for other purposes such as verifying that an application is installed correctly with the correct permissions, or getting a list of all the files that were installed as part of a package.

Matt Hansbury from MITRE asked: What is more useful to the community? What *was* installed or what *is* installed? Or are both useful?

Comment from a participant: It's useful to have historical data, for instance in case residual files are left behind. It may be worth having two tests, one using the pkgutil package receipt database for historical information and one using the System Profiler for current information.

Comment from another participant: Historical information does not seem very useful vs. current information. Also, can the historical information be trusted as accurate?

Shawn Geddis from Apple said that pkgutil provides only history of applications installed using Apple's Installer. Not all applications use Apple's Installer, so information on those would not be included.

A participant suggested that since it sounds like we can't trust pkgutil historical information to be accurate and complete. Instead, we should just use System Profiler to gather currently installed application information.

Jasen Jacobsen from MITRE observed that pkgutil –verify didn't appear to complain about anything even after a package was manually deleted.

A participant suggested that if the community decides to include a pkgutil test, maybe call it something like Apple installer history to avoid confusion about the information that it provides.

Conclusion

We will explore creation of tests that leverage the System Profiler to provide live data. The utility of the proposed pkgutil_test is questionable. At the very least, the pkgutil_test should not go in an OVAL release alone at this time, but potentially could be included alongside System Profiler tests.

Further discussion on this topic is required to achieve a consensus on how to proceed with regards to checking Mac OS X packages. This discussion can occur over the oval-developer-list.

Solaris Schema Updates

Dan Haynes
MITRE

Introduction

This talk's primary goal was to review the updates proposed in the OVAL Sandbox in support of Solaris 11. Warren Belfer of Oracle introduced the new updates in the Solaris 11 operating system and was followed by Dan Haynes of MITRE presenting the supporting OVAL tests that have been added to the Sandbox.

Problem

The most interesting addition to the new version of Solaris for this group is that it now ships with OpenSCAP. While there is not much content currently, there is a roadmap for including several hundred OVAL checks. Another addition to Solaris 11 is the Image Packaging System (IPS), a package management system that provides a framework for the management of installations, upgrades, and removals of software packages. Most services are now managed by Service Management Facility (SMF), which is a management feature to ensure essential application and system services run continuously despite hardware or software failures. Config files in “/etc” are no longer the recommended mechanism for configuration, though they do remain for backwards compatibility. Their use is discouraged. The primary configuration properties are all available in the SMF repository. The automated installer is available as an XML manifest and profile to perform the install. With Solaris 11, a user is able to use IPS to use boot environments, which would provide for a method to reboot into a previous environment if the package upgrade did not fulfill expectations. The boot environments are represented as deltas from past environments to maintain a small footprint.

With the introduction of IPS and configuration management with SMF, some tests within the current Solaris schema have been superseded. Additional tests are required to fully utilize the new functionality provided by the new IPS and SMF utilities.

Proposal

To address some of these new features of Solaris 11 relating to the packaging system and SMF, several new tests have been proposed. The proposed tests are:

- *package511_test* - Check the metadata of installed packages. Uniquely identify a package using the FMRI value.
- *packageavoidlist_test* - Check which packages have been flagged to avoid from installation on the system. Retrieves the FMRI values of packages in this list.
- *packagefreezelist_test* - Check the packages that have been frozen at a particular version. Retrieves the FMRI values of packages in this list.
- *packagepublisher_test* - Check metadata of package publishers configured on the system. Identify the Object based on the name, the type (“origin” or “mirror”), and the origin URI.
- *image_test* - Check the properties associated with an image. Each image is identified by a path on the file system. Currently designed to work for one image/property value pair.
- *facet_test* - Check the values of facets for the specified image. Similar to image_test, but specifies a facet value.
- *variant_test* - Check the values of variants for the specified image. Similar to image_test but specifies an architecture variant for an image.
- *smfproperty_test* - Check the values of properties associated with an SMF service. Retrieves service, property, FMRI, and value for a specified service.

For each Test, Danny demonstrated the value it provides and presented use cases to support the additions. He also provided sample content to show how these would appear in an OVAL Definition file. There were several open questions of formatting decisions on how to represent some entities.

Discussion

Throughout the discussion of the new Tests, several audience members raised questions about technical details regarding implementation. During the introduction of the package511_test, one audience member was concerned with whether package lists were obtainable from a non-global zone. Warren Belfer responded that he was unsure if this was possible as most of their testing had been performed in the global zone. He added that to get the full list of packages, you typically needed to be within the global zone. Another audience member was interested in whether there would be a packageverify_test, analogous to the rpmverify_test. It was determined that this request made sense as there were similar tests for other platforms, however no effort has been performed towards this goal yet.

Regarding the image_test, there was a question about what would be returned for several cases. Primarily the audience member was concerned whether a user would be able to specify a “nil” value on the name entity or utilize pattern matching to collect all the properties of an image. As the test was designed, these options would both be possible. However, an interpreter would return multiple Items with one property each. It was to be determined whether a record datatype would better suit this formatting need.

One question posed by Danny to the audience was whether it was better design to keep the FMRI as one string while specifying Object entities in the smfproperty_test, or to decompose that value into its subcomponents. The audience was favorable of consistency in the schemas, so since the FMRI was specified in the service portion then it made sense to maintain that style. It was explained that this question arose from considering how to iterate over properties where they could all be named differently. A member of the audience asked for clarification of whether it would confuse an interpreter by specifying a property name in the FMRI. Danny responded that by the test’s design, it was specified that the property is not to be included in the FMRI entity. Such a case should report an error if encountered.

Lastly, a general consensus of the Solaris 11 schema updates was gathered by the host. It was seen by the audience that with direct participation of a primary source vendor verifying these methods of collection, that these schemas would make a good addition to the OVAL Language.

Conclusion

As mentioned by one member of the audience, having the primary source vendor working with MITRE on the creation of the platform extensions creates the highest quality outcomes. Having authoritative sources confirm the design and operation of their platform’s tests is ideal for OVAL. A general consensus was that these tests would greatly benefit OVAL by expanding the Language without modifying any core elements. Some questions regarding formatting issues and how to best represent the structure of collected Items still remain to be discussed.

Apple Platform Discussion

Shawn Geddis and Josh Wisenbaker
Apple

Introduction

This session discussed Apple OS X and iOS platform configuration topics, with the goal of increasing the security automation community's Apple platform knowledge.

Background

Apple has provided CVEs for years and is working on stepping up its community involvement in other security automation areas. Apple has started the SCAP-on-Apple open source project on Mac OS Forge: http://scap-on-apple.macosforge.org/

Presentation

Defaults domains are used to store preference information. The defaults command line tool should be used to access this information, rather than reading or writing any underlying plist files. The defaults command line tool changes the defaults databases which then change the underlying plist files. The plist files may not always be in XML format but may sometimes be in binary or JSON format (the plutil command line tool can be used to convert plist files between XML, binary, and JSON formats). The defaults information can also be accessed programmatically through Apple's APIs.

Managed Preferences were used on previous versions of OS X but are now deprecated starting with version 10.7.

Configuration Profiles originated on iOS but are now available in OS X as well. Configuration Profiles are the preferred mechanism for new developments on OS X and are the only option on iOS. Configuration Profiles are preferably managed through Apple's Mobile Device Management (MDM) protocol, which is now not just for iOS but for newer versions of OS X as well. The MDM protocol can be used to silently manage the installation of Configuration Profiles. Configuration Profiles also can be manually installed by double-clicking the profile in the Finder, can be delivered over email, can be managed through shell scripts, or other related techniques.

Configuration Profile controls can include account, VPN, and Wi-Fi information, policies (such as passcode length requirements), restrictions (available functionality), and other settings such as PKI certificate auto-enrollment.

Several tools exist for creating Configuration Profiles:

- Profiler Manager – Part of OS X Server (available in the Mac App Store for only $19.99, used to be much more expensive)
- Apple Configurator – Designed for iOS, can't handle OS X specific features
- iPhone Configuration Utility – Available for Windows (unlike the others), but its features are getting stale
- Manually edit the Configuration Profile plist file with a text editor

Configuration Profiles can generally be shared between OS X and iOS platforms. However, some specifics may vary due to differing functionality.

To verify that a Configuration Profile is in place on a system, use the profiles command, system_profiler command, the System Preferences GUI, or use the MDM protocol. If a Configuration Profile is installed, then that should provide confidence that its policies are in place and are being enforced.

On iOS, third-party applications are prevented from having access to system-wide device settings and status information. The MDM protocol is the only way to programmatically access this kind of information on iOS.

Registered Apple developers can access the Configuration Profile and MDM protocol reference specifications at http://developer.apple.com/.

The SCAP-on-Apple open source project was started to encourage close collaboration between Apple and the security automation community. It has involvement from Apple, NSA, NIST, and others. Apple is providing dedicated engineering resources to help create security automation content for OS X and iOS.

Discussion

Question: If plist files are manually modified, will that create a conflict with Configuration Profile settings?

Answer: The Configuration Profile setting should still try to win. If that's not the case there may be a bug.

Conclusion

Apple provided a wealth of information to help the security automation community better understand configuration functionality of the OS X and iOS platforms. Apple will participate through the SCAP-on-Apple forum to help security automation efforts.

OVAL Adoption Conversation

Omar Santos
Cisco

Introduction

Omar Santos of Cisco led a discussion about OVAL Adoption, citing Cisco's experiences with OVAL, and then leading a discussion about adoption. The intent was to generate conversation and ideas about how the adoption of the OVAL Language and tools could be expanded beyond where it currently stands. In particular, Omar provided perspective from a primary source vendor.

Cisco's Experiences

Omar began by describing his experiences with OVAL at Cisco. Omar's role is that of a security expert and as such does not directly bring in revenue for the company. His job is to manage product vulnerabilities and as customers began to more and more ask about security, it became feasible to convince the company to begin to invest in OVAL and OVAL content. Cisco has not only been working on updating and expanding its related schemas in the OVAL Language, but has also begun publishing advisories in OVAL for Cisco's IOS platform.

Omar shared some findings about Cisco's experience with OVAL. He mentioned that while the OVAL content that they were providing was heavily downloaded, it wasn't clear that it was downloaded by Cisco customers in great enough volume. Many of them were from students and other non-OVAL end users. Additionally, Cisco already provided the same type of information that was now being published in OVAL, in an enterprise solution. Given that these products still make money for the company, convincing them to switch to using OVAL is challenging without customer demand. Lastly, because OVAL can be complex, integrating with existing products is challenging.

Discussion

Following his experiences, Omar opened up the floor for discussion, the following points were made:

Others in attendance agreed that they saw this type of difficulty in adoption. It was also suggested that since OVAL has historically focused on the desktop, the group might need to continue to look at other vendors, outside the desktop space.

Another significant barrier to wider OVAL adoption is lack of content. Without a greater availability of content, end users are less likely to demand OVAL support. It was pointed out that the more that primary source vendors publish their security advisories and other authoritative content in OVAL, the more likely the demand is to rise.

Another pointed out that the non-security sections of corporations often have a difficult time conceptualizing the value of OVAL support. Additionally, some agreed with Omar's suggestion that OVAL's complexity is hurting adoption. A follow on comment was made that the ideal driver of user adoption would be full clarity and disclosure as to how much time and money is spent on doing non-OVAL/non-automated solutions to these problems. If the users knew more about this cost, they would demand something like OVAL, whether or not they ever actually saw any XML or not.

Conclusion

The great discussion led by Omar presented a few key findings:

1. The best driver of additional OVAL adoption and support is end user demand and clear business reasons. More clarity and disclosure of current non-OVAL or non-automated solutions to security issues might help further solidify user support.
2. Larger amounts of content will help drive further vendor and end user support of OVAL. Additional primary source vendor participation helps with this issue.

The complexity and depth of OVAL can be a hurdle to adoption. More shared content and better content authoring tools may help with this.

Juniper JunOS Recap

David Solin
jOVAL

Introduction

David Solin of jOVAL began by reviewing the proposed JunOS schema, originally presented during last year's Developer Days event. He then updated the group on networking device compliance in general as well as on the status of Juniper with regards to OVAL.

During the session a number of discussions were held both in regards to specific JunOS schema tests as well as broader networking and primary source vendor support questions.

Problem

Over time OVAL has progressed from a very Windows-centric set of schemas to a far more broad set, including Linux, UNIX, Mac OS, etc. It continues to expand into yet other platforms, thereby increasing the number and diversity of the endpoints about which it can make assertions. One of these expanding platform sets is networked devices, such as routers, switches, etc.

Currently there exists schema to support some Cisco devices, but support is lacking for Juniper devices, which makes up a significant amount of networked endpoints in both government and commercial enterprises. This section discussed the effort to create a schema that can support the assessment of these Juniper devices.

Additional Background

David also added some additional background, mentioning that Cisco has started not only directly supporting its OVAL schema and expanding it, but also publishing their security guidance as OVAL content.

Juniper is also now publishing security advisories on the web, although not in OVAL or SCAP format. They have also created a Puppet client for JunOS that can assess a Juniper device against an expected state.

Proposal

The proposal presented by David includes four tests, inspired by similar tests found in the Cisco schema. It was noted that no new enumeration, data types, or other global/core change to OVAL was required for the new schema. The proposed tests are:

- *global_test* - The global test tests for the existence of a specific line in the configuration file.
- *line_test* - The line test is used to check the properties lines output by using a SHOW command.
- *xmlline_test* - The xmlline test is used the same way that a line test is, but operates against an XML version of the line.
- *version_test* - The version test provides the ability to make assertions about the version for various Juniper devices.

David discussed the various details for each test, including showing sample content for each. Additionally, he presented a set of open questions, intended to foster conversation, captured in the following section.

All of these tests are available in the OVAL Sandbox as part of the proposed JunOS schema.

Discussion

Following his walk through the new proposed tests for the JunOS schema, David led a discussion starting with a couple of open questions for the group.

The first question was directed at DISA specifically, asking if any representatives from DISA felt that this schema and its related tests were necessary to support STIGs. At least one DISA representative suggested that DISA would like to be able to use these types of tests in STIGs.

He then also asked if the group felt like any tests were missing from the proposed schema and specifically if the DISA Level 3 STIG could be expressed using the proposed schema. The general consensus here was that the schema, as it has been proposed, would be sufficient for the task. It was also pointed out that NETCONF might be able to provide some of this functionality, should OVAL support a NETCONF test, as suggested in past presentations. There was some discussion about whether OVAL should add NETCONF tests to augment the platform-specific schemas like Cisco IOS and Juniper JunOS. In general it seemed that for some cases, the platform-agnostic nature of NETCONF would be advantageous, while in other cases, the platform-specific tests might be required.

Following David's specific questions, a few other questions were fielded. It was suggested that Juniper, as the primary source vendor, would be the best suited to create and maintain the schemas as well as the content for Juniper devices. The group agreed that this would be ideal, and explicitly asked the Juniper representatives about this topic. The representatives pointed out that while in general they support the ideas presented, they could not speak for Juniper and therefore could not commit to such support.

The general consensus was that while Juniper's support and leadership here would be preferred, in the absence of such support, it was best to create a first attempt at a Juniper schema for the Language to allow for the interrogation of such devices.

Others asked about how these networked devices would be interrogated. David pointed out that jOVAL could be used in local, offline, or remote modes and that the schema does nothing to preclude any of these styles of data collection. Another participant pointed out that while STIG support is important, organizations will also want to be able to create their own Juniper-related policies.

Finally, an attendee asked for more clarification on the difference between using the proposed tests and the existing textfilecontent_test. David suggested that since the textfilecontent_test requires a path, its usage doesn't easily support this case and therefore specific tests, such as the ones proposed, fill that role more effectively. Lastly, it was pointed out that we could consider re-working or re-thinking the file type tests (textfilecontent_test, file_test, xmlfilecontent_test, etc.) such that they were able to work against a blob of text collected from an arbitrary construct.

Conclusion

From David’s presentation, it was clear that the community sees a need for support for Juniper devices both within DISA for the STIG work, and also within the larger community. Juniper seems to be lightly supporting this work, with some indication that in the future more support would be possible. The group seemed to agree that while the preference was for Juniper to lead and take responsibility for its OVAL schema and content, in the meantime moving forward with the JunOS schema, as it is currently proposed, is preferable to the current state with limited ability to interrogate these devices.

It was also clear from the comments and discussion during this section that there needs to be further revisions to the schema in order to prepare it for inclusion in a new version of the Language. This additional discussion should take place on the OVAL mailing lists.

Mobile Platform Discussion

Mike Peck
MITRE

Introduction

Mobile devices are everywhere with over 900 million Android devices activated and 600 million iPhone devices sold. Given this, there is a strong desire to use these mobile devices securely in the enterprise. However, there are numerous threats against mobile devices including both traditional computing threats as well as new threats which target their portability, connectivity, sensors, and limited resources.

Some specific threats include the device being lost or stolen, untrusted connections, and malicious applications that utilize device capabilities in a negative way. To counteract these threats, you may want to enable hardware encryption, make sure only authorized applications are installed, check applications and their permission requests, or set up remote wipe capabilities in the case that the device is lost forever. Lastly there are unpatched vulnerabilities which can be exploited. This is further complicated because Google pushes out patches, but, it doesn’t control what gets installed on a device and it is left to phone vendors and carriers to deliver those updates. Therefore, it is important for enterprises to know if their devices are unpatched and make sure any available patches get applied.

Problem

The main goal is to secure mobile devices. Mobile device management systems can enforce policies on mobile devices and monitor their compliance status, however, they often utilize proprietary policies and checks and there is no standard way to express compliance policies across all vendors or represent the results of compliance checks limiting the usefulness of these results within the larger enterprise security (e.g. for dashboards, access control, etc.). Furthermore, there is a reliance on manual compliance checks which does not scale well to an enterprise environment.

OVAL helps with this problem because it can provide a mechanism by which enterprises' can come up with OVAL Definitions that check for compliance with their pre-defined policies, load them into their mobile device management systems which know how to talk with their devices, and then can produce results in a common format that can be integrated into their enterprise security infrastructure. Having properly configured and monitored mobile devices will significantly reduce the security risk from these threats.

Proposal

In general, for Android, mobile device management vendors will ship an agent application which gets installed and knows the Android API calls to gather the useful information. It also knows how to use the device administration feature on Android where it requests special permissions from the user and it can set the password complexity policy or make sure device encryption is enabled among other things. The management application can also check for compliance in other areas although there is no way for Android to stop users from downloading and installing applications and there is no policy to enforce that, but, it can monitor the list of all applications, check that USB debugging mode is disabled, or WiFi settings are configured properly, etc. So, while it can't stop the user from modifying the device, it can detect when misconfigurations happen.

Another feature in the Android schema is the appmanager_test which gets all of the installed applications on the device. Unlike Windows or Mac OS, on Android and iOS there is plenty of structure to how applications are installed and when you query the list of applications you will get an accurate list. One can gather the name, version, permissions, and contents of the PKI certificate that was used to verify the package signature among many other things.

There are also device settings such as USB debugging or device administration permissions that can be checked. For example one can check for applications installed from places other than the Google Play store, that some management application is installed and has the device administration permissions so a remote wipe of the device is possible, and that no unauthorized apps have this permission. Password checks can also be written to confirm that the device is configured according to policy.

Lastly, you can collect various pieces of operating system information such as version and manufacturer which will let you check if any known vulnerabilities are applicable to the device.

There is an experimental Android schema available in the OVAL Sandbox which is based on a contribution from SecPod last year and has since had new features added. A proof of concept application has also been built. The application uses OVAL Definitions to determine which system characteristics information to collect and generates the OVAL System Characteristics XML which is then shipped off the device where it can be evaluated by an OVAL interpreter. The application source code is also available in the OVAL Sandbox. There are also a few OVAL Definitions, based on DISA's mobile device SRG, in the OVAL Sandbox. An OVAL Definition that checks to see if all applications are signed by an authorized key was presented.

There are a few open issues with regards to Android support. First, what else should be added to the schema? Second, how can differences between Android versions be handled as each official release often introduces new policy capabilities? The final issue is that some vendors have included their own extensions beyond what is provided in stock Android. How should this be handled? One approach would be to have individual schemas for each vendor and another approach would be to only add capabilities when they are added to stock Android.

Apple provides a mobile device management protocol which can be leveraged to get things like serial numbers, versions, installed applications, check if hardware encryption is enabled, etc. There are many other settings and information that you can retrieve and these settings can be configured with configuration profiles. A proof of concept has also been developed for this, though is not out on the website yet, but, it demonstrates how iOS with OVAL could be supported. The proof of concept involved developing an OVAL System Characteristics schema for iOS and having the Intrepidus MDM server dump out its information in this format. Another approach to support iOS would be to write OVAL Definitions that check the plist files passed back and forth by the MDM protocol. In either case, hopefully there is the potential for alignment between Mac OS and iOS.

Discussion

Question: Are there any plans for integration with SEforAndroid policy definitions?

Response: SEforAndroid is a research project out of NSA and is not currently in any official Android release. However, some vendors have included it on their devices. I would like to add support for SEforAndroid in the Android schemas, but, I am waiting for it to show up in an official Android release so that I will know what the changes should look like. Rumor has it that Android 4.3 will get announced and will have support for SEforAndroid. If that is the case, we could add support for things like making sure the policies are in place. If there are any other ideas for capabilities to add, please let us know.

Statement: You have to go way back to find those devices in production. iPhone 3GS was the first to have hardware encryption capabilities.

Statement: Probably the easiest thing to do is to have the MDM server do its thing and simply query the databases for the device characteristics information. For example, the MDM I set up this morning has a Postgres database behind it that I can just ask and do a query on the database to get the state of the device.

Response: Yes, that makes a lot of sense for a proof-of-concept. I did not have a real MDM to work with. So, I think that you would be leaning towards Option 1 to query the database and then dump out the OVAL System Characteristics XML?

Response: As long as you have access to that data, you can put it in a usable form rather than trying to insert something into the protocol. Also, the protocol is documented.

Statement: The folks that you said had a python MDM server may just be using the code that we handed to them at the Apple Developers Conference.

Statement: We are fairly open about how the protocol formatting works.

Response: That makes sense to me and the part that we want to standardize is not how do we actually get the information, but rather, once you have the information, how do you format it and what do you do with it.

Question: Since there are different layers of the OS stack where you can get data, should there be some notion of trustworthiness depending on where we get the data and are there some sources that are more trustworthy than others? Have you considered this?

Response: So, I am not sure that is in scope from an OVAL perspective, but, certainly you would want to make sure that there isn't any malware that is making it lie about its status information. This is a problem with mobile device management whether it is standardized or not. The smart charger that David Weinstein created is one way of determining trustworthiness. While your phone is charging, it would put the phone in a special state, from a trust perspective, and it would make sure that there are no malware rootkits interfering with it. Also, hardware root of trust is showing up on mobile devices and having some special area, where code can run, where it cannot be interfered with malware and you can select integrity information from there.

Question: For Android, how did we get the information?

Response: We wrote an application that runs on the device and the application knows the Android SDK calls to grab the required information and it outputs the OVAL System Characteristics XML file.

Question: Do you think Android is basically a JVM on top of Linux and that a definition can actually extract information from the operating system?

Response: The application can do whatever you want it to do making the Android Java calls. If it needs to go under the hood for some reason, you could include native code in the application where you could execute system commands. I don't think we have had to do this yet, but, you could if you needed to. So, this is the proof of concept in an enterprise environment. A mobile device management solution would come with agent software that could scan the device. We would hope that MDM vendors would incorporate this into their existing way of doing things. So, the agent would grab the information from the device, it goes into some database on the back end, then you can grab the OVAL System Characteristics out of there. Since we didn't have that, we wrote our own application to run on the device and gather that information. There are plenty of different ways to get the information, but, that part doesn't matter so much.

Question: How does the application inspect the device to see what applications are installed and how they are configured?

Response: Any application on Android can use the package manager class to interrogate all of the other applications on the device and make calls for the name of the application, the permissions it has requested, and who signed it along with many other things. On Apple iOS, an application can't inspect the state of other applications, but, you can use the MDM protocol to get that kind of information.

Question: Have you talked to any MDM vendors?

Response: Not really. I mentioned this to a few vendors and got some head nods, but, I think we need to talk with the government folks to help convince them.

Question: Have you looked into Microsoft devices and do they support OVAL or device management?

Response: I haven't looked much into Microsoft or BlackBerry as I have been focused on Android and iOS. However, we should consider looking into those platforms as well. Also, if there is some alignment between Microsoft phones and their desktop operating system, hopefully the same types of practices can be used between the platforms.

Question: DISA put out a mobile STIG and right now it is just XCCDF which is supposed to be executed manually. Are you going to look into how certain functions of an MDM can be made part of the automation? Also, DISA takes about 18 months to create a STIG, is this going to be a problem for mobile devices?

Response: I know DISA is working on the speed of the mobile device STIG process. For example, the Samsung KNOX STIG was released before Samsung KNOX devices were ready for purchase. I would like to encourage DISA to include OVAL Definitions along with their STIGs when the time is right. We will try to go through the KNOX STIG and come up with some examples of how to automate those checks because right now it just has manual compliance checks. There is also no support for it in the OVAL Language which makes it a chicken-and-egg problem, but, that can be addressed.

Question: Have you implemented all of the constructs in the Android schema in the proof of concept application?

Response: Yes. The proof of concept application implements everything in the schema so far. This should be helpful for anyone trying to implement the schema. Any vendor can grab the application and see how we are querying the information on the system and that is the same type of information that vendors would want to gather with their agent applications.

Conclusion

Overall, there seemed to be some support in favor of adding the new Android schemas. There needs to be more discussion regarding iOS.

INI Test

William Munyan
Center for Internet Security

Introduction

INI is a structured, text-based configuration format leveraged by numerous applications and operating systems, such as PHP, Microsoft Windows, and MIT Kerberos. This presentation proposed an OVAL schema extension to improve OVAL's ability to examine INI files.

Problem

INI files are widely used on Windows to store preference and configuration information. They are structured files with a defined syntax. Currently the only way to examine these files with OVAL is to use the textfilecontent54 test. This is insufficient for capturing or testing the context-sensitive contents of INI files. Definitions built using the textfilecontent54 test use complex regular expressions and variables, making the creation of these definitions complicated and error prone.

Bill also presented content samples, showing how one would write an INI test with the textfilecontent54 test and one with the proposed INI test, clearly displaying how the new test removes a very complicated regular expression and replaces it with a clear and easily understood piece of content.

Proposal

The proposal is to add an INI file specific extension to OVAL for inclusion in the 5.11 release of the language. The ini_object is comprised of the following components:

- filepath
- section
- subsection
- name
- instance

The ini_state and ini_item would add the value component.

This proposed test and related artifacts can be found in the OVAL Sandbox.

Discussion

The presenter raised the question of whether the ini_object should mirror the other file-based objects and support path and filename components. Consensus was that these components should be included.

OVAL - XCCDF Evaluation Reference Architecture

David Ries
jOVAL

Introduction

XCCDF provides a way to encode security policy in a standardized format and can include external checks to determine adherence to the defined policy. Historically, the policy rules were largely represented as OVAL checks. Vendors over the years have learned a great deal about the coordination between the XCCDF policy and the checking language. While OVAL had long been the sole expressly defined checking language, with SCAP 1.2, the Open Checklist Interactive Language (OCIL) has also been added as another checking language (used for non-automated checks). Additionally, some vendors have made use of non-standard checking languages. With this knowledge, this section offered up some recommendations for handling OCIL, OVAL, and other checking languages.

Problem

By experimenting with incorporating a different checking system such as OCIL, some problems were discovered. It seems to be assumed that an SCAP bundle will be executed locally on an endpoint and OCIL questions will be directed to a person on the machine in a synchronous fashion. It is expected that one person will provide answers for all questions. There are routing challenges to get the question to the most appropriate person, and dealing with questions where the answer may change over time.

Another problem is that OCIL Questionnaires are more costly to query people where issues of timing and repeated queries for humans come into play.

Proposal

David has proposed a ticketing system to alleviate some of the issues outlined above. This system should be able to delegate a question amongst an organizational chart, and handle responsibilities for answering in a timely manner. The topography for this solution is similar to the SACM Working Group's proposed architecture. The controller would coordinate the request on behalf of the endpoint with automated checks being executed on the endpoint while the manual checks would be entered into the ticketing system. The ticketing system would create the survey to query the designated survey respondent and manage handling the results, notifications, escalations, and timeouts. The controller would then store all the system results and handle them as new information becomes available. Upon completion, this would be dispatched to the XCCDF evaluator to collate results to determine policy compliance.

While one might argue that it is not the XCCDF specification's issue to include such routing and organizational metadata, amending the specification to include this type of information could also provide value. This would need to be done in a way to allow for metadata sufficient to enable the routing of such a ticketing system. The metadata element is not required for the specification and is referred to as including the "best practice" since the XCCDF. It would be more helpful to become an actual element. Adding this would add a lot of value to evaluation efforts.

Discussion

During the introduction of David Ries' solution making use of a ticketing system for questionnaire management, one audience member shared his anecdotal experience. The audience member explained that the ticketing method is not commonly used amongst IA engineers. While it could be implemented, it introduced an added expense. That audience member had likewise built an OCIL manager and had the same issues. His solution was to use email, which could incorporate OCIL using HTTP and accept an HTTP response. Instead of a ticketing system, it was people on staff responsible for delegation of duties. They would be able to know who was available and who was on vacation, which the ticketing system would not. David acknowledged that his approach was sensible, yet would still benefit from a ticketing system with the acceptance of the added cost.

One audience member was concerned with the use of an OCIL tool designating people to answer a question multiple times over a span of systems. He wished to see a case that when the answer would not change, the question may be asked once then replicated. The response was that with the right group policies, this can be done by one person on behalf of all machines for which it applies. It was noted that when determining which machines may be grouped in such a fashion, it could still require a manual effort to link those clusters. A few audience members were also concerned with the frequency of which a single question was asked. They shared their views that the question should only be asked as often as the answer may change. It was their experience that spamming OCIL questions made it more likely to be ignored, which defeats the automation of the check. In addition, one audience member pointed out that the age of the question is likely to affect evaluation as well.

Another audience member had raised the question of using XCCDF to incorporate metadata to provide questionnaire routing and scheduling information to the OCIL manager. It was not believed possible to use tailoring to provide this information in the current schema. This was met with the follow-up question of why the metadata element was not moved into the check system. One member of the audience who participated in the drafting of the XCCDF schema responded that at the time, that was the only place to put the metadata.

A member of the audience shared their view on how they viewed collection in an asset-centric way. Should one asset be moved to another location, then the maximum age for a question could dramatically change the calculated risk. The presenter responded that different policies should be drafted based on the location of the asset.

Lastly, a discussion was held about the ability to make changes to the schema to achieve all these ideas. One person noted that it was known to take months to get a new feature implemented. They viewed the standard as being in a stagnant position. Given that one working group desires to move the standards to an international forum, he was concerned that it is not a complete and fully operational standard where it would be ready to get moved there. It was noted that for the time being, the XCCDF specification is published as a NIST IR and falls under NIST ownership. The hypothetical question was posed that should this standard be taken to an international forum, what would be the timeframe and time duration for making changes, and who would have control. An audience member familiar with IETF procedures responded that no specifications have been submitted to the IETF yet. Down the road, if the document gets accepted into the IETF, then anyone can request changes. If the consensus of the working group is there, then the changes would be made. He noted that in order to quickly make changes without IETF approval, extensibility points should be included in the draft. He added that it could take a long time to change an RFC if you have not put in the extensibility points. In that case, then it could take a year to get a change.

Conclusion

There was general consensus that the inclusion of OCIL checks caused some issues with XCCDF evaluation. Not all audience members were convinced that a ticketing system was the most effective method. However, there were examples given of this being used elsewhere. David Ries made good suggestions for what would be required moving forward with XCCDF and OCIL integration, which led to good discussions about making these desired changes