EMAP Developer Days, August 29-30, 2011
Minutes

All presentations are posted at http://scap.nist.gov/events/2011/emapdd/presentations/index.html. These minutes do not duplicate the content of the presentations, but instead focus on capturing discussions involving the direction of EMAP and the individual specifications being covered.

Welcome and EMAP Program Overview, Paul Cichonski, NIST

Debug levels were discussed. The decision was made a while back to have debug out of scope for EMAP because of its level of complexity. Different people have different notions of what "debug messages" are. One comment was that many debug messages are so implementation-dependent that it would be extremely difficult to standardize parsing them, other than standardizing an encapsulation method. It might make more sense to just try to classify what each debug message means in general. On the other hand, debug messages may be good indicators of attack attempts, such as processes crashing.

There was also brief discussion of other types of events that might or might not be in scope for EMAP, such as system status messages.

Common Event Expression (CEE), Bill Heinbockel and Tom Graves, MITRE

CEE Overview

There was an extended discussion of what events are, whether certain types of things are events or not (status, debug, and error messages, for example), and what CEE's scope should be. Some types of things (such as status messages) might be much easier to include in CEE than others (debug) that need models. The concept of CEE being scoped for "action events", composed of subject/action/object, was raised. There is concern about having separate protocols for different event/message types because all of these events/messages are in a single file; systems aggregate these things together. It might make sense for CEE to have a way of distinguishing event types.

There was general consensus that terminology is very tricky in this domain and that agreeing on common meanings for words like "event" is important.

It was unclear if alerts were events or not and whether they should be in CEE's scope. Although alerts are often targeted at people, aggregated, and used more for continuous monitoring purposes than event management, they shouldn't be neglected for this type of work. It might be fundamentally different for a third-party system to generate alerts or correlated events instead of dealing with raw event data.

There was discussion of what the goals of CEE and other similar efforts are—are they trying to change industry best practices with regards to how they log? One response was that the initial goal is thought to be normalization of terminology and representation. It's not possible in the short term to change how logging is done; the long term goal is better policies and directives. Someone else commented that standardizing semantics will change the industry.

Event Modeling

The next major discussion topic was field names. Having rich field names would allow products to parse names without having their direct definitions. However, the proliferation of field names is a concern, especially since there is a desire to allow communities to define their own field names because of the huge range of events. CEE cannot eliminate all use of conflicting terms, just minimize it. CEE wants to maintain backward and forward compatibility; no name will be changed after being issued (other than being retired). Changing CEE too quickly would mean that vendors won't be able to keep up.

There are two possible scenarios for field names: either have structured field names or introduce structure into the actual event representation. Some vendors and logging folks have issues with the latter approach. Should the field name be somewhat duplicative of the event content? Or should the location of fields within an event dictate part of their meaning? The latter means that you have to keep track of a lot of things to have the necessary context.

Event Language

Someone asked about the status of CEE, particularly timeframes and mobility support. Regarding timeframes, CEE 0.6 was just released, and MITRE wants community review and feedback. Hopefully CEE 1.0 will be released some time in 2012. Regarding mobility, that hasn't been looked at yet, but there's nothing inherent in CEE that prevents it from being used in mobile devices. Most mobile platforms already support XML and JSON. However, there could be size issues. The main barrier is getting representatives together to engage that community.

There are other size limit concerns. Sometimes an event needs to capture additional data, such as a form or a network packet stream. These are huge, well above size limits—how do you handle this? It is important to keep events and data together, but data is supplemental to the event. Trackback URLs, out of band delivery mechanisms, and packaging (e.g., SOAP envelopes) are possibilities. ARF is handling this through URL references and XLink. For CEE, people need some way of linking to external data.

The first CEE fields are specified differently from the other fields in the proposed XML. This takes away some of how they could be managed in the CEE Profile. They could be specified just like all the other fields are instead.

Event Comprehension and Analysis

Someone noted that the profile capability is a model constraint use case, not unique to CEE. There may already be standardized approaches for this that avoid defining another one. So far, a suitable approach has not been found, but if one is found it could definitely be adopted. The intent of the profiles is primarily to provide guidance—a common set of representative CEE fields—and requirements and recommendations to vendors about what data fields and events should be logged. If there is a conflict, such as one policy requiring that something be logged and another policy requiring that the same thing not be logged, will CEE handle this? A tool could be developed to identify conflicts and allow users to resolve them.

Log consumers can map to a profile through an event ID. For example, Microsoft has their own event IDs. Each ID is unique for an event type. Event profiles would be written based on those IDs. Someone asked if the event record has a mapping to an event profile; this is not necessary, but helpful. It would help to have the CPE in the event.

Some products can change what they emit, such as capturing greater detail under certain circumstances. An event record will tell you what information is available, while a profile will tell you what the product

is capable of emitting. It's important to know what the capabilities were, and what was meant when an event was emitted. Someone thought that this was a far-off scenario (10 years), while someone else commented that we need to look forward to provide the foundation today to allow this to be developed tomorrow.

Someone asked about any work on prototyping, reference implementations, etc. There has been discussion with various vendors about doing these types of things, but the status of these efforts is unknown. Someone else commented that until we start coding around examples, we are not going to be sure what works and what doesn't. There is a need to bring together communities and build profiles. Next year there will be a big focus on building all types of profiles.

Sharing CEE Events

There was a discussion of CLT conformance levels, with someone noting that most of the level 1 requirements could be handled by SSL. Extra requirements could be moved to level 2 and then level 1 could be handled just with SSL; some of these requirements are only for the highest-security environments and shouldn't be below level 2. Someone else commented about having a negotiation feature in level 0 or 1. It might also be good to have layers of protocols instead of putting everything together; this would be a more modular and flexible approach.

The Open Group Distributed Audit Services (XDAS) v2, David Corlette, Novell

Someone asked about differentiating success and accept statuses. Accept: policy allows it, success: operation worked.

There was brief discussion of how models (XDAS, CEE, etc.) could be compared and contrasted—code both? Integrating them? At some point people will need to look at the specs side by side and figure out all the mappings. There is also a need to have a unified requirements list; if CEE and XDAS are being designed to achieve the same thing, then duplication of effort should be reduced. Vendors should talk through this and develop unique and unified requirement lists. Someone else commented that the CEE and XDAS efforts have models that line up very closely and that we are focusing too much on acronyms.

Standardizing Event Parsing and Translation, Paul Cichonski, NIST and George Saylor, G2

Someone noted that a single CPE may have one or more logs and one or more log formats. There are often one-to-many mappings from CPE to logs; for example, a single product may record an event in multiple logs.

There was a question about putting the conditional logic in the input processing instead of output. This is a good idea and a viable option.

The real difficulty in event parsing and translation is having a structured format, particularly for handling multi-line events. Doing the actual conversion to CEE or XDAS is not hard. We need to focus on mapping to some model. Another consideration is that most products allow you to choose what types of events are logged, but very few allow you to change what fields are logged for each event. Also, localization may affect whitespace and other characteristics of event records.

Multi-line event records were discussed. Do multiple lines turn into one higher-level event? Can we distinguish low-level and high-level events? If we keep multiple lines as multiple "events", can we correlate them after the fact? In some cases we can link the lines to each other even if they are split up,

but in other cases the lines must be captured together and in sequence. One idea is to inject an event ID during parsing of multi-line log entries. However, this gets complicated when multiple lines are interleaved between events. Another way to think of this is that a carriage return isn't always a record delimiter, and that some multi-line log entries have a standard format while others don't.

Another topic of discussion was the use of XML versus regular expressions. Regexes will comprise far fewer lines than XML. Tools could take XML and process it into something more efficient. There was disagreement as to whether writing XML or regexes would be more time-consuming. There was also disagreement about the value of having XML and a translation (logic) model, when people could just write parsers (implementations) and share them within the community. NIST has been responding to SIEM vendor feedback, saying that they don't want to be writing their own parsers and that there's no standardized way to exchange information. There was clarification that what's being discussed is processing existing log sources, not getting raw data through development of drivers, etc. Also, it's not necessary to write XML content for every possible format—maybe just write them for the top 50? Fundamentally, at least 90% of the work is figuring out what the events mean.

There was additional discussion of whether a logical model was needed. Someone asked how you make sure that different tools exhibit the same behavior and map to the same CEEs without a model. It was also noted that XML isn't necessarily the best way to represent semantics. Not having a model will harm correlation across products. We want output to be deterministic and consistent, with the same core—same code, same ruleset, etc. Some people want the transform from A to B to be more flexible, and to have a generic language to define a variety of transforms. Other people feel that this is yet another layer and unnecessary.

Regarding a possible reference implementation, it would be impossible to support all source formats. As a stopgap measure, we could select a subset of well-used, required source formats. Someone else commented that many existing products in this space only handle about 20% of the events today.

There was a question about whether there is a market for OEEL; maybe products don't really need it. One person commented that there isn't a market, but people will expect these capabilities to be there. We are complicating the issue—we should make sure the output is acceptable without dictating exactly how it's generated. Vendors are already doing a lot of transforms, for a variety of reasons. An alternative is to create a controlled vocabulary and put it into a PDF; however, if that vocabulary is well controlled, then it can probably be machine readable instead of a PDF.

General consensus was to look into both options. Neither one is optimal, and neither one is going to succeed as is. We need to think about what the factors are: operationalization, maintenance costs, etc.

Standardizing Event Rule Languages, Paul Cichonski, NIST and George Saylor, G2

There was discussion of existing rule languages. Many products already have proprietary rule languages. Someone asked if a rule language could be used to do filtering.

Cyber Observables and Integration with EMAP, Sean Barnum, MITRE

A few people noted that WMI overlaps with the DMTF CIM model.

There was an extended discussion about real time versus non-real time (forensic) detection. Ultimately a network activity model is needed to improve decision making and interoperability. The proposed model allows logic at multiple levels.

We need to be thinking 10 years out, where we want things to be, instead of looking 2 years out. It will take 4 to 5 years to get this into the field, and another 5 years to mature it. Things need to be moving to the endpoints.

There was concern about the malware use case. It was clarified that the intention isn't to use this model for regular, comprehensive malware detection purposes, but rather to look for other systems that have been affected by the same malware that has already compromised systems. This is a big use case for information sharing between agencies, especially for zero-day threats and forensics purposes.

Someone asked about the CybOX relationship to IODEF. IODEF would likely be a wrapper around CybOX.

Integrating Event Management and Continuous Monitoring, Dave Waltermire, NIST

There is a need to formulate strategies at the top level and push tactical actions down to lower levels. This is a content driven approach to CM. Organizations can create new content as needs evolve and push that content down to sensors. The hope is to support both real-time and batch collection of data. We may need to explore some new workflows for real-time activity.

Someone asked about deconflicting content. With this model, the sequence is orchestrated; Analysis/Scoring does the deconfliction, such as choosing a preferred data source over others when conflicts occur. There were also concerns about efficiency, and in the proposed model there would be a task manager, a human-oriented component for scheduling queries, etc. You can also define digital policy for when these things can occur, such as requiring human approval for certain types of queries.

Built in to all of these models is the capability to handle situations where host X can contact host Y, but host Y cannot contact host X because of security policies—restrictions on data flow, etc. The models need to be able to support these real world scenarios, with high interoperability but with respect for network security controls, etc.

Vendors are doing these CM function today, but in proprietary ways, and not broken into modules.