

The Open Group / XDAS v2

Brief History and Progress on v2

David Corlette

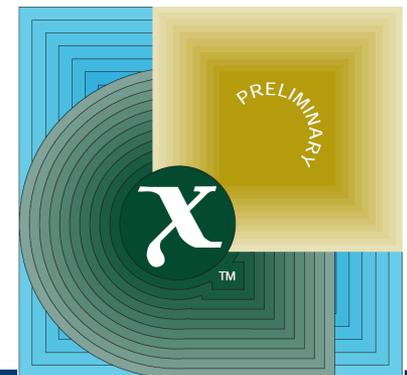
Product Line Lead

DCorlette@novell.com



Brief History

- XDAS (X/Open Distributed Audit Service) was a draft standard published in 1997 by The Open Group
- To the best of our knowledge, was and is the only open eventing standard that:
 - Is aimed at a general IT security and compliance domain (as opposed to a specific subset such as vulnerabilities)
 - Defines a clean high-level data model that maps to almost any interaction of IT resources
 - Defines both a syntax and semantics for expressing complex interactions beyond just “timestamp, host” (as with syslog)
 - Defines a taxonomy of categories for events that maps well to the types of questions that CxOs and auditors ask



Brief History, Part II

- XDAS v1 includes very classic C-driven APIs for fetching and filtering events.
- The OpenXDAS project created an open implementation of the standard (C library).
- Not widely adopted beyond a few reference implementations, hence it never achieved formal standard status.
- In 200?, Novell approached TOG with the idea of updating the XDAS standard.
 - Remove the API dependencies and focus on syntax/semantics
 - Update the object model to include new types of systems

Current Status

- The XDAS draft document has been proceeding slowly; a current draft which describes the ideas mentioned here was released in November.
- There are major new ideas in the draft, but we need more input, validation, and detail.
- Heavy activity in coordination work with other organizations: DMTF, NIST, and MITRE
- Activity mostly on hold as DMTF gathers cloud requirements – these will flow back to XDAS
- I'll present the major problems we've faced along with our current solutions.

Key Problems

Problem #1: Vendor Bloat

- Every vendor believes that their system is special and unique and does things no one else does.
- This triggers an insistence on vendor extensibility, “profiles,” and other bloat features that make the standard complicated.
- **Consumers** should dictate:
 - Requirements that the standard must implement
 - How vendors are allowed to extend the standard
 - What “conformance” to the standard means

Solution: Use Consumer Sources

- The Open Group has relaxed its normal rules for group participation.
- Compliance Requirements
 - PCI, NIST SPs, SOX, etc
- Consumer-generated use cases
- SIEM use cases
- Keep it simple
- Tie events to user actions where possible
- Re-use existing successful standards
- Extensibility only at “lower” levels

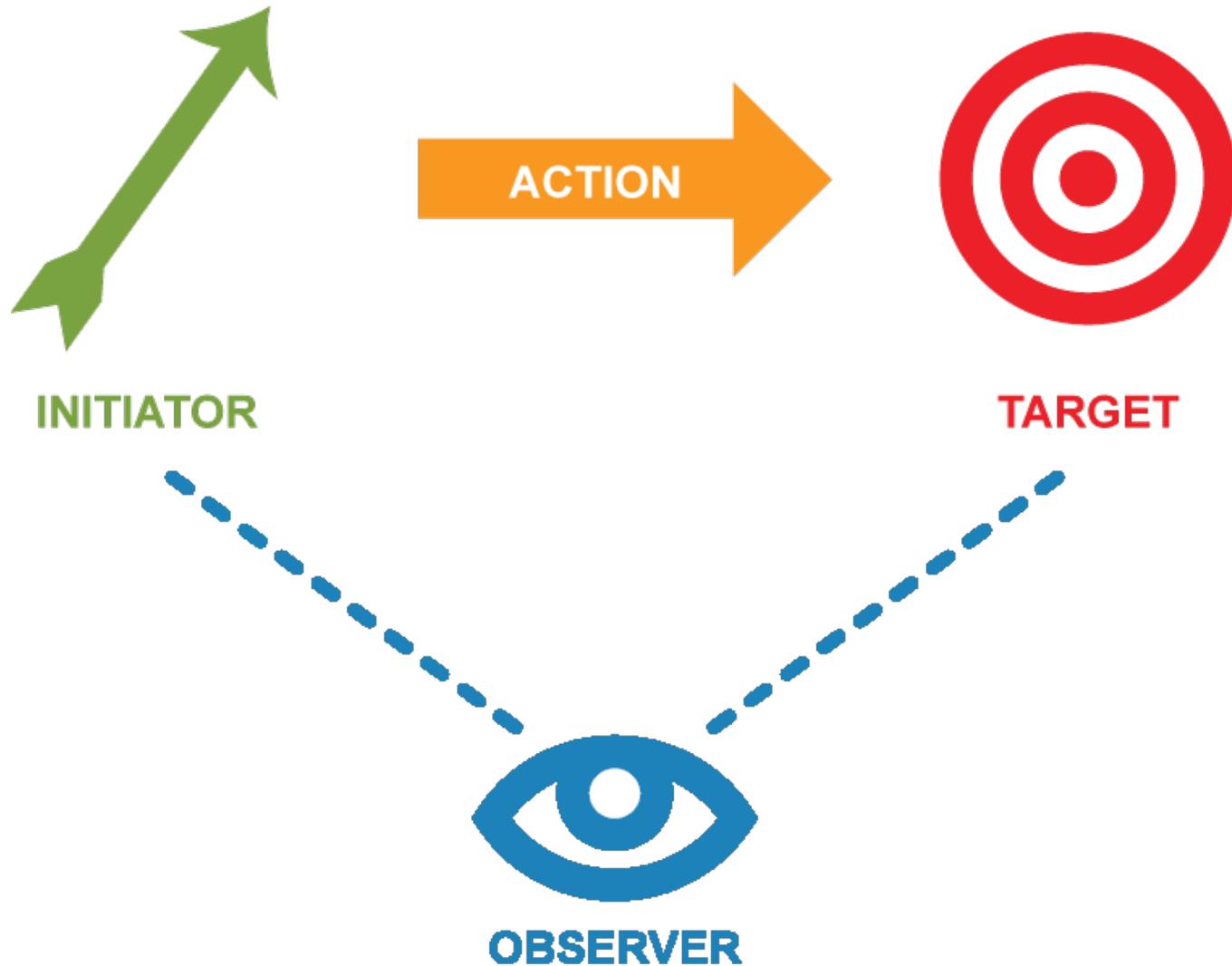
Compliance Example

- 10.2 Implement automated audit trails for all system components to reconstruct the following events:
 - 10.2.1 All individual accesses to cardholder data
 - 10.2.2 All actions taken by any individual with root or administrative privileges
 - 10.2.3 Access to all audit trails
 - 10.2.4 Invalid logical access attempts
 - 10.2.5 Use of identification and authentication mechanisms
 - 10.2.6 Initialization of the audit logs
 - 10.2.7 Creation and deletion of system-level objects

Problem: Event Model

- Need a consistent, high-level event model to which we can map event information
- Other event models have been very domain-specific and/or complex
- Can constrain our scope somewhat (initially) by focusing on “security” events, which implies intentional action
 - Ends up excluding things like variable state reporting, debug logs, etc
- Leverage concepts from XDAS v1

Solution: XDAS v2 Domain Model



Problem: Object Model

- Next we need to figure out how to describe what *kind* of Observer, Initiator, Target
- Only a few possible Initiators (people, services, hosts) and Observers (services, hosts), but many many kinds of Targets (accounts, hosts, files, ...)
- Needs to be able to describe relationships (service ON host, account IN domain, etc) and can't be constrained by number (group ADDED TO group – this is not the same as “deleted 10 files”)
- Other models (including SIEM products) have tried to come up with a list of fields – the list gets big
- Don't want to re-invent the wheel!

Solution: Common Information Model

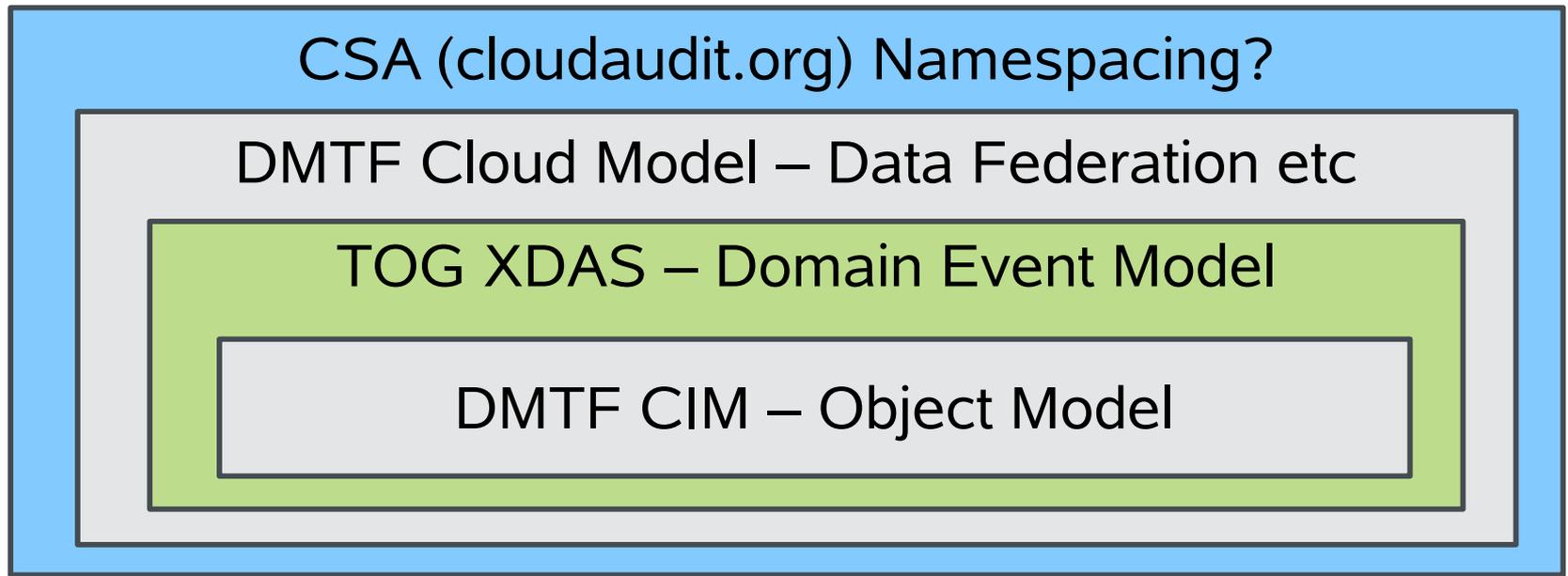
- DMTF CIM model has already done a lot of work describing all “manageable” resources in IT – these are the things one can “act” on
- Inheritance, classes, references are all covered
- Caveat is that the model is large, and contains lots of irrelevant data – we will scope it down as part of XDAS (for example, Initiator can ONLY be Account, ProtocolEndpoint, SoftwareFeature, or ComputerSystem)

Account
Name
Host[]
OU[]

IPProtocolEndpoint
IPv4Address

Benefit: DMTF Coordination

- Proposed use of CIM has led to interaction with “Cloud Audit, Data Federation Working Group” in DMTF

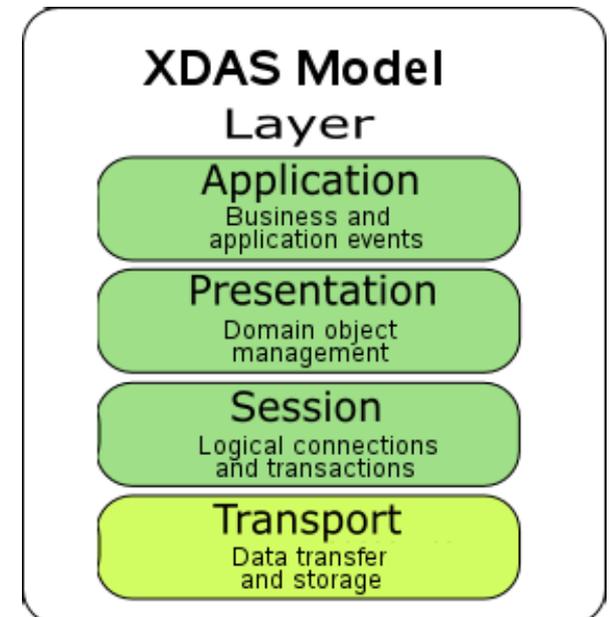


Problem: What Event Am I?

- Consider an activity where `/etc/passwd` is modified: this is clearly an “file modify” - but it's also “account modify”
- Current models don't provide support or guidance on *what to report*, they simply provide the structure
- This leads to confusion and major differences between what different products report, even if the same format is used

Solution: Define Event “Layers”

- Inspired by OSI, but more by analogy and for terminology than strict correspondence
- Applications should report events at the highest “layer” they know about.
 - 'adduser' should report 'Account Modify'
 - FS driver should report 'File Modify'



Benefit: Clarifies Extensibility

- A side benefit of this approach is that it helps guide when vendors can/should extend the taxonomy
- Lower layers should be more obvious and common with events like CREATE, DELETE, START, STOP, and hence should rarely need extension
- Upper layers have application-specific events, so may need extension more frequently (may be room for profiles in this domain)
- Event consumers can more easily guarantee support for key lower-layer events

Problem: Who Am I Related To?

- Virtually all IT activity is transactional in nature, but these transactions can span services and systems
- There is a major attribution problem – consider a scenario where a user logs into a web service that uses a single proxy account to access data in a backend database – how can we know who requested the data?
- It's very hard to force the web service to pass info to the backend, or vice versa
- This just gets worse in the cloud!

Solution: Correlation IDs

- XDAS will define two correlation IDs:
 - **Grouping ID** : groups related events together *at the same level* – typically reported by a single service
 - **Transaction ID** : connects “request” events to “result” events
 - These don't always need to be explicitly passed from caller to callee – could be determined by session properties such as connection endpoints

App1
Events



App2
Events

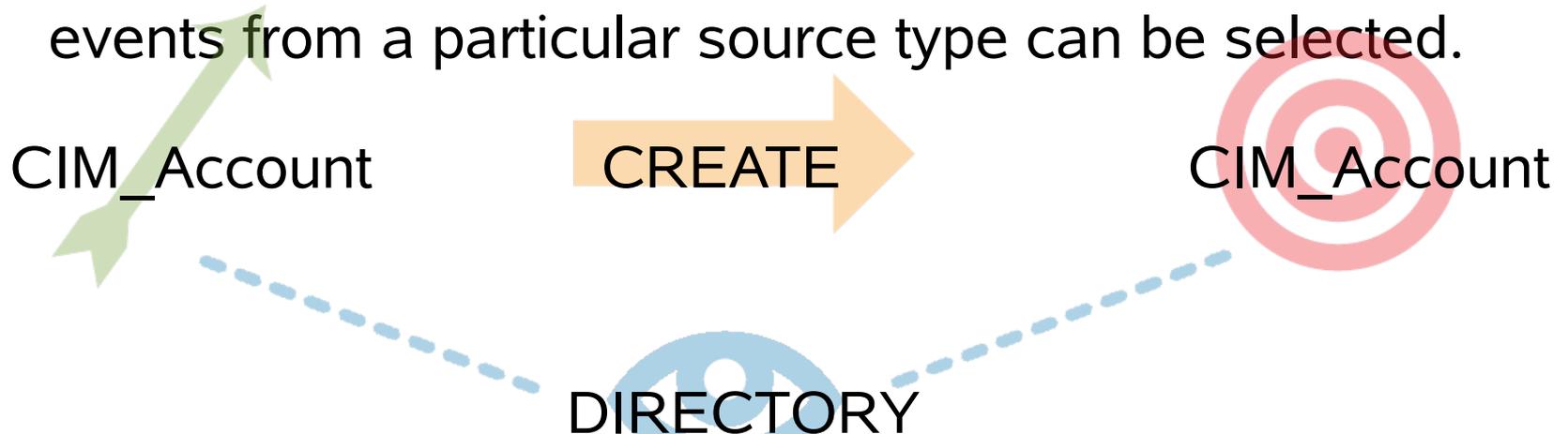


Problem: Where Am I?

- One of the key use cases we need to achieve is to provide ways to classify events so that events of a particular type can be found easily.
- XDASv1 used a sort of hybrid taxonomic classification that mixed the action that was taking place with a descriptor of the type of target that was acted on
- Other schemes use tags and/or domain-specific techniques that can make it hard to meet general compliance use cases:
 - Show me all blocked connections from my firewall
 - Which user accounts were created yesterday?

Solution: Multiple Taxonomies

- XDAS v2 strictly segregates the Action taxonomy (the “verb”) from anything else.
- As with XDAS v1, a separate Outcome taxonomy will also be used to report the result/status.
- The Target is classified by virtue of being a CIM object or a related object.
- The Observer will also be classified so that sets of events from a particular source type can be selected.



Distributed Management Task Force Cloud Auditing, Data Federation

DMTF CADF

- Formed as a sub-group to the Cloud Management Working Group (CMWG)
- Intends to define specifications as to how “cloud” event data can be federated to consumers (customers, other cloud vendors, etc)
- Is defining a data syntax/model and requirements for the interfaces, but the actual interfaces will be defined by CMWG

CADF and XDAS/CEE/?

- CADF intends to define “mappings” to one or more event syntax standards such as XDAS and/or CEE
- The Open Group has an Alliance Partner relationship with DMTF (as does the Cloud Security Alliance) but MITRE does not
- CADF has looked at the work going on within XDAS, CEE, and CSA closely and will be attempting to influence development in same

CEE and XDAS

Coverage

- XDAS is focused solely on the event record contents; CEE covers transport, filtering, etc.
- Our initial hope was that MITRE would leverage XDAS for the CEE syntax piece, and focus on the other pieces such as transport. This has not come to pass.

Hierarchical Schema

- XDAS has selected a hierarchical schema, whereas CEE has selected a flat schema
- The flat schema is easier to represent in “NVP” forms
- Our belief is that the hierarchical schema provides additional information that is lost with the flat schema, and is more naturally extensible (e.g. subclassing).
- Our belief is also that the hierarchical schema will perform better...

Hierarchy of Data Relevance

XDAS events will “surface” the critical data to the top of the record; systems worried about speed can simply drop the “lower level” data.

```
<XDAS action="CREATE" outcome="SUCCESS"  
initiator="Account.Name='Administrator'"  
target="Account.Name='dcorlette'"  
observer="ComputerSystem.Name='HR_SRV1'">
```

```
  <Initiator Domain="NOVELL" Name="Administrator" Id="0"/>
```

```
  <Target Domain="NOVELL" Name="dcorlette" Id="1002">
```

```
    <Roles Role1="Users" Role2="HR" .../>
```

```
  </Target>
```

```
</XDAS>
```

Taxonomy of Action

- XDAS uses a hierarchical taxonomy of event action, outcome, and object classification; CEE uses simple tags and flat schema
- Our belief is that the hierarchy is important because it allows grouping of related events:
 - Data object
 - Database table
 - Row 15
 - Account
 - Service account
 - Denial
 - Invalid credentials
 - Bad password
- The top level(s) will be fixed and non-extensible, but lower levels may support vendor extensions

Profiles

- CEE supports profiles that can redefine any level of the event architecture; XDAS will restrict extensibility to areas that provide additional detail to the “core” event structure.
- Our belief is that this will protect and enforce the consumer's ability to define simple queries, since it should prevent vendors from redefining what an “authentication” event looks like at the highest level.
- We also believe that profiles make implementations much more difficult, especially in a distributed environment like the cloud where the consumer may not even know what vendors/products are in use (and that information might be proprietary).

Standards Efforts

- The Open Group / DMTF are “official” standards organizations; MITRE is not
- Unclear how this will play out
- Want to avoid a holy war and discuss the technical differences between the standards (w.r.t. how well they implement the requirements)
- Ideal is that we align the two standards as closely as possible (given different target “markets”) and parent standards (CADF, EMAP) can consume either

Conclusion

Significant Inhibitors

- There are major issues to be worked out between the two “standards tracks” – namely TOG/DMTF and MITRE/NIST – that are currently underway
- Some of these issues are political, some technical, some due to differing “markets” for the standards under development
- Right now the set of real event consumers involved in the standards efforts is quite low, and the pros/cons of any particular technical decision are not well stated in terms of how that decision will affect consumers

On the other hand...

- There is clearly a groundswell of interest in a real event standard, although mostly from the vendor side to date
- Many of the fundamental concepts are well aligned across the different efforts, and similar/parallel solutions have been proposed

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. NetIQ Corporation may make improvements in or changes to the software described in this document at any time.

Copyright © 2011 NetIQ Corporation. All rights reserved.

ActiveAudit, ActiveView, Aegis, AppManager, Change Administrator, Change Guardian, Compliance Suite, the cube logo design, Directory and Resource Administrator, Directory Security Administrator, Domain Migration Administrator, Exchange Administrator, File Security Administrator, Group Policy Administrator, Group Policy Guardian, Group Policy Suite, IntelliPolicy, Knowledge Scripts, NetConnect, NetIQ, the NetIQ logo, PSAudit, PSDetect, PSPasswordManager, PSSecure, Secure Configuration Manager, Security Administration Suite, Security Manager, Server Consolidator, VigilEnt, and Vivinet are trademarks or registered trademarks of NetIQ Corporation or its subsidiaries in the United States.

