14th International Command & Control Research & Technology Symposium – C2 and Agility

Title: Protecting Identifiers in Cross-Domain Environments

Topics 10: Collaborative Technologies for Network-Centric Operations

Submitted by: Dr. Sam Chamberlain
U.S. Army Research Laboratory
In Support of The Joint Staff / J-8 / MASO
ATTN: AMSRD-ARL-CI-CT
APG, MD 21005-5067
410-278-8948; Fax: 4988
sam.chamberlain@us.army.mil
chambesc@js.pentagon.mil

This Page Left Intentionally Blank

I-138: Protecting Identifiers in Cross-Domain Environments

Dr. Sam Chamberlain

U.S. Army Research Laboratory
In Support of The Joint Staff / J-8 / MASO
ATTN: AMSRD-ARL-CI-IC
Aberdeen Proving Ground, MD 21005-5067
410-278-8948
sam.chamberlain@us.army.mil, or
samuel.chamberlain@js.pentagon.mil

Abstract

Unique identification of objects and their associated data representations have received significant attention in the past 10 years. Developing an efficient identifier allocation and tracking scheme that transparently spans security domains requires finesse. It is not uncommon for information to be created in a lower security domain and copied to a higher domain. The rigor by which the data is maintained varies widely, as does the resulting difficulty in maintaining consistency of the data and its identifiers. But identifier uniqueness and traceability is not the biggest concern. In the age of the Internet, it is easy to pull together disparate pieces of information to build a picture not intended for public release. Previous practices such as data masking are no longer satisfactory. It is easy to believe that because an identifier is an unintelligent number that it can be passed around without compromise. This paper will describe the policy and technical logic behind a policy of managing identifiers and presents the argument that identifiers, even unintelligent ones, must be treated with the same care as the data they identify.

1. Introduction

The Global Force Management Data Initiative (GFM DI) reached a major objective by achieving initial operating capability of a suite[1] of information sources called *organization servers*[2] (OS) that provide access to default organizational and forces structure data for the Department of Defense (DOD). This data is produced and maintained by the agencies across the DOD who are responsible for creating it; consequently, there are currently seven servers in development: one by each Service, one by the Joint Staff that includes the combatant command headquarters, and two by the Office of the Secretary of Defense that includes a special OS to handle the needs of the subset of organizations that make up the DOD intelligence community.

A significant property of the GFM DI force structure data is the ubiquitous use of unique identifiers. Every piece of data in the OSs is associated with a unique identifier. The GFM XSD

[1] Suite: a group of software programs sold as a unit and usually designed to work together.
[2] The term "server" is used in its original meaning: a software application program that accepts connections based upon a request / response paradigm. In this usage, it does not mean a physical computer system.

was developed from an information exchange data model that was based upon a relational database model. Consequently, it reflects the properties of normalization that breaks up the data into atomic groups so that, among other properties, the data only appears once in the data schema. While this may be viewed as a nuisance to some, it has beneficial properties for data management that extends to security purposes as well. The GFM XML schema definition (XSD) includes 12 primary elements whose identifiers are known collectively as the set of Force Management Identifiers, or FMIDs. One FMID that is used to identify organizational elements, or OEs, is also called the Organizational Unique Identifier, or OUID. Every element of data in the GFM XSD is associated with an FMID.

Although most force structure data is unclassified (but often considered sensitive), there will be portions of the organization structure that are classified. The approach for handling force structure data across security domains is to create the data at its appropriate (lowest) classification level and then duplicate the data to higher classification environments where information may be added at the higher classification. This process can occur across as many boundaries as necessary and ensures that identifiers remain consistent across security domains. This propagation is illustrated in **Figure 1**. Other resources also ensure that the identifiers remain trackable as they move up the domains even though they were created in a different domain. Thus, in the GFM OSs, common data remains commonly identified and trackable across security domains.

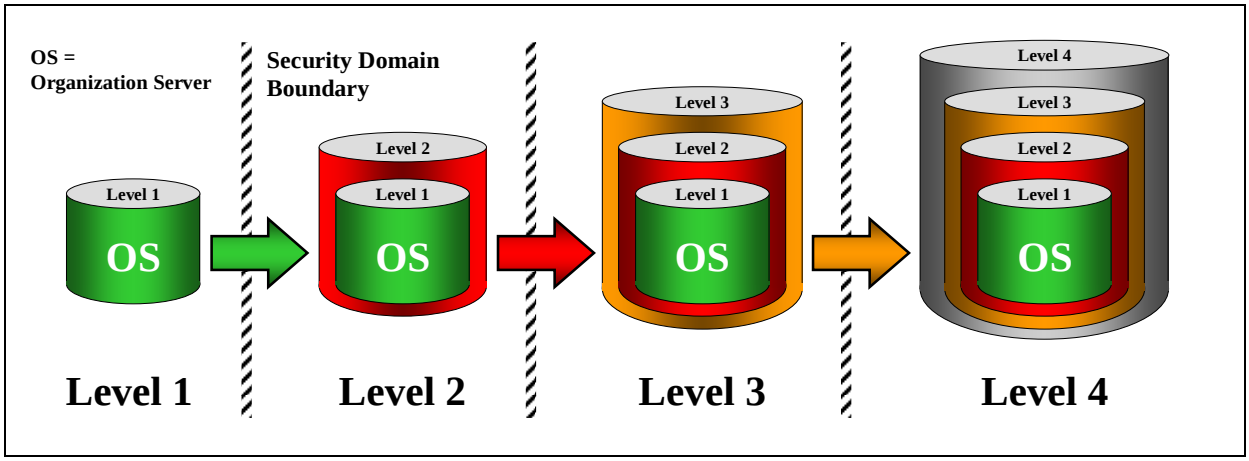

Figure 1: Data Propagation Up Security Echelons

2. Classification Scope – Data Entities, Attributes and Values

Entities and attributes are fundamental building blocks for data schemas. It is common to consider entities (or objects, the two terms are used interchangeably in this paper) as abstract containers defined by a set of attributes. An important distinction is the difference between an attribute name and value. In the simplest terms, an attribute is a placeholder that describes a characteristic or property of an entity or object of which it is a part. An attribute has a name and a value. This is illustrated in **Figure 2** with an object (or entity) template on the left and an instance of the object on the right. A simple example is the entity named *platform* with an attribute named (maximum) *Speed*. One can see the distinction between the attribute name "Platform.Speed" and its value "Platform.Speed=1000 MPH." Another attributes is named "Platform.ID" and it contains an

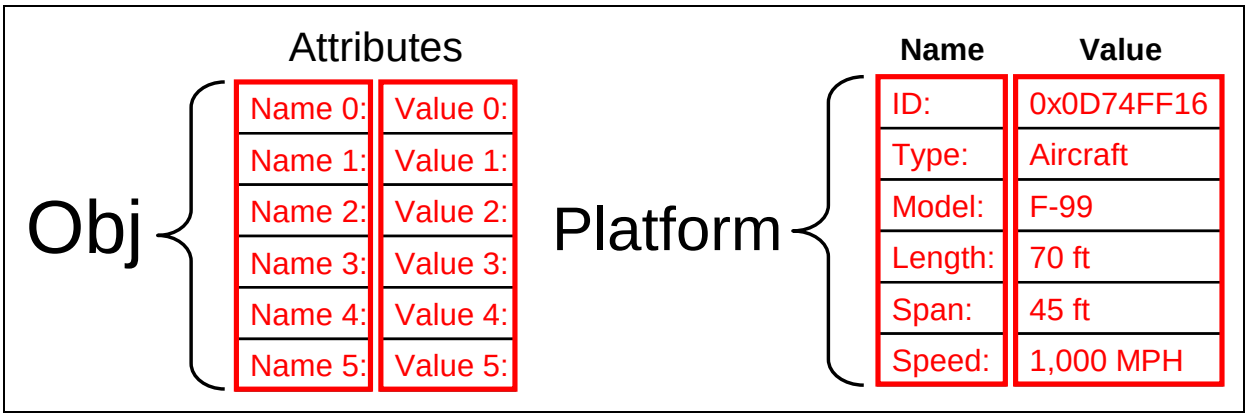

Figure 2: Entities Have Attributes with Two Parts – a Name and a Value.

identifier for the entity. In the GFM XSD, FMIDs are defined as the set of (12) attribute names that contain the values of the unique identifiers for the 12 basic GFM entities.

When describing the classification of data, one property is its "atomicity boundary." This refers to the resolution to which the data can be identified as classified. There are two general cases:

Case 1: Attribute resolution: one or more specific attributes of the entity can be identified as having values whose sensitivity is higher than the other attributes in the same entity, thus making the classification of the entity that of the highly sensitive attribute.

Case 2: Entity resolution: the existence of entity is to be hidden; therefore, its classification can not be ascribed to any specific attribute, but only to the entity as a whole.

Using **Figure 2**, an example of Case 2 world be if the existence of the F-99 aircraft was classified. In this case, removing any of its attributes will not change this fact and lower the classification. There is no way to identify a particular attribute as being the reason for the classification. This is the case often encountered with the GFM DI force structure data where the existence of particular organizational elements is classified. An example of Case 1 would be if a particular property of the F-99 aircraft were classified, say its maximum speed, while its existence is known. Therefore, removing this attribute would lower the classification of the data. Alternatively, it might be allowable to change the actual value (1000 MPH) to another value, like > 200 MPH or just the word "classified." This is a trivial case of the practice of data masking. However, it is also plausible that the fact that there is an attribute named "Speed" might in itself provide insight that is not desired. When this is the case, only removing the attribute from the data will produce a lower classification. While these cases may seem obvious, understanding this distinction is important when defining the interaction of data with varying protection levels.

The definition of atomicity boundary becomes more intricate when unique identifiers are incorporated, especially when these are single attribute, unintelligent identifiers (unintelligent meaning that no information can be gleaned about the entity from the identifier). Of particular interest is when the identifier is unique over a wide scope, commonly called an enterprise-wide identifier. There are many advantages to having such "globally" unique identifiers because they allow one to easily and unambiguously narrow the reference to the data to a single attribute. Although the tendency is to consider the unique identifier as just one of an entity's several

attribute, it is special because, unlike the other attributes, it is synonymous with the entity itself (and its set of attributes). This makes an attribute that provides enterprise-wide identification special because of its power to provide extreme referential convenience that can be used for positive and negative purposes.

3. Unique Identifiers as Attributes

Figure 3 contains an entity X (on the left) with six attributes, one of which is named ID with value "0x0D74FF16" that is an enterprise-wide, unique identifier for the entity. As just stated, this property makes this attribute special because it is guaranteed to uniquely identify the entity within a wide data domain, and is therefore synonymous with entity X; this property is not inherently true for any of the other attributes. Because of this property, entity X (composed of n attributes) can be easily decomposed into n-1 sub-entities (one for each of the n-1 attributes that is not the unique identifier) using the unique identifier attribute as an aggregator. In other words, entity X can be arbitrarily decomposed into sub-parts, or conversely, its sub-parts can be aggregated using the common, unique identifier. The sub-parts can be grouped in any combination. This practice is found frequently in data schemas. Often (to save space) the set of attributes for an object varies based upon the content of the object. For example, a *platform* object may represent a ship or an aircraft, but it will include a *draft* attribute only if it is a ship, and a *ceiling* attribute only if it is an aircraft. Therefore, the set of attributes can differ based upon the properties of the entity, but the identifier attribute retains its common name and form. In relational data models, this occurs in a structure called a generalization hierarchy. The point of this example is that the elements of an entity are often distributed across a data schema (as illustrated by elements x_1 - x_5 in **Figure 3**) and do not have to reside in one place.

If an entity is classified because of its existence (i.e., Case 2: entity resolution) then every part of the entity, regardless of how its data elements are distributed, must have the same classification.

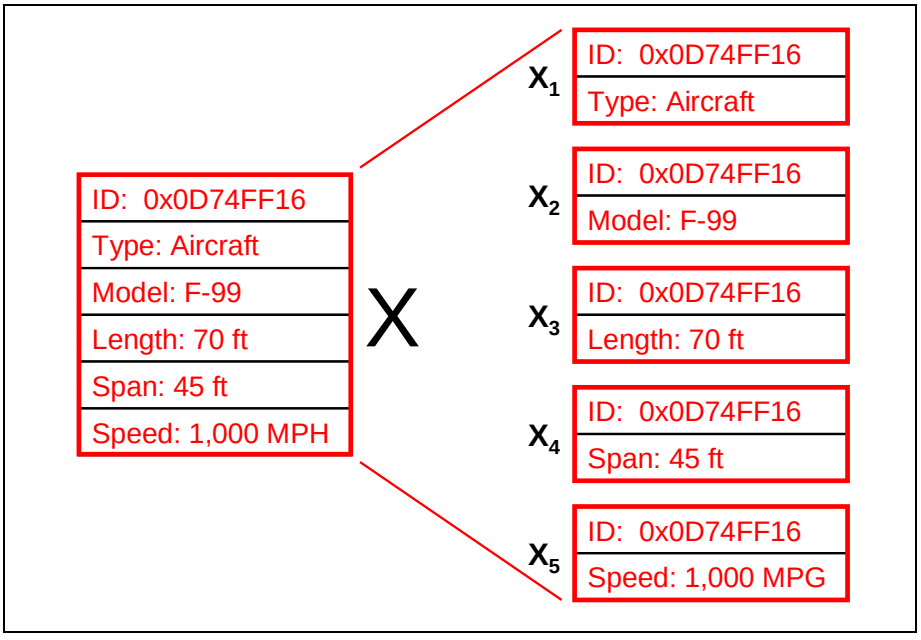

Figure 3: Unique Identifier Equivalence with an Entity and a Group of Attributes.

(Recall that this case implies that there is no subset of attributes that can be determined to be the cause of the assigned classification level.) Clearly, the reverse of the decomposition of an entity into subparts using its unique identifier can be considered an equivalent problem from an aggregation perspective. The five entities on the right in **Figure 3** (x_1 - x_5) can be trivially aggregated into the single entity on the left (X).

The reason for this discussion is to show that, in cases where the existence of an entity is the cause of its classification, *the unique identifier (intelligent or not) must be classified at the same level as the data it identifies*. Simply isolating the identifier from the rest of the data is not sufficient to consider it at a lower protection level because the identifier is synonymous with the entity and can be used to coalesce related attributes into the entity regardless of how they are distributed in the data schema. Because of the identifier's universal scope, this is true whether the data resides in a single source or across the Internet. In other words, in **Figure 3**, the classification of sub-entities x_1 - x_5 must be the same as the classification of entity X. Further, all six attributes, must also be considered of the same classification, including the unique identifier (which seems intuitively logical since it is synonymous with the entity it identifies).

To reiterate a previous (probably obvious) point, if an entity has been assigned some classification level, its attributes and their associated values have the same classification. A value without an association with an attribute name is not classified. For example, the number 6 can not be classified, but the attribute Platform.Quantity=6 can be. The same is true about unique identifiers. When it is stipulated that a unique identifier must be classified at the same level as the data it identifies, this does not refer to the isolated value, but only when the value is part of an attribute. However, if a set of values is provided as the result of another process, such as a query, then the values would be classified at the level of the highest attribute. This is because the query criteria provide an aggregation function (commensurate with an attribute name) that resulted in a related set of data. This is an important point that must be regarded when dealing values that superficially appear as isolated but in reality are not.

4. Identifiers as Imported Attributes

Identifiers are commonly used as imported attributes between entities in data schemas. Called Foreign Keys in the relational database model, this occurs when a unique identifier (e.g., primary key) is used in one entity to reference another. In normalized data models, this is a common occurrence. Overall, relating entities is a complex subject, so this discussion is restricted to the effects caused by interaction between entities of different classification levels that may cause a higher classification level to be propagated to the importing entity.

Recall Case 1 (attribute resolution) where one or more attributes of an entity can be identified that cause the overall classification level of the entity to be raised because of them. In normalized data, such as that on which the GFM XSD was based, imported attributes are frequent and classification levels must be adjusted to protect entities at the correct level. It is common for an entity that provides a connection between two other entities to have its classification level raised to the highest level of the entities involved. The tenet stipulated in the previous section, that a unique identifier must have the same classification level as the data it identifies, makes this process uncomplicated and intuitive.

Consider the situation illustrated in **Figure 4**. Entities X and Y represent organizations (i.e., X.Type=Y.Type=Org) and entity Z represents a command and support relationship between them (e.g., Z.Type=Association). Thus, entity Z represents that one org is the parent of the other via some relationship type. The association is implemented by importing the ID attributes from each of the org entities, one into the parent (Par) attribute and one into the child (Child) attribute.

The entities X, Y, and Z can have one of several classification levels as maintained in the attribute labeled Class (because the attribute Class applies to the whole entity, it may be considered mete-data). There is no inherent classification resulting from the entity or attribute names; therefore,

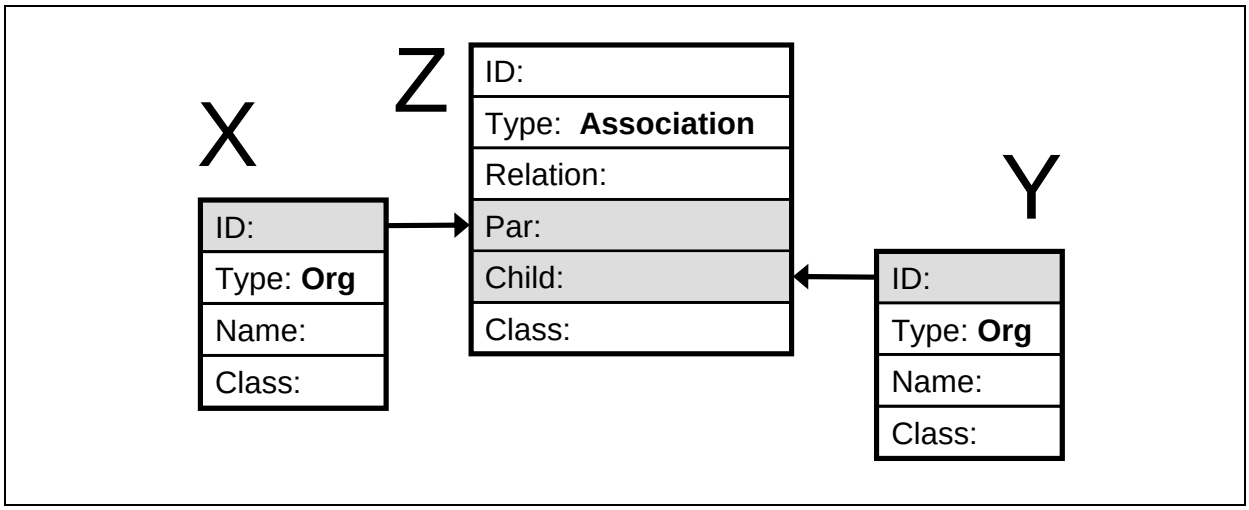

Figure 4: Associations Via Imported Attributes

the classification level of the entity is based solely on the attribute values. Either Case 1 or Case 2 may apply; that is, the entities classification may be based upon a particular attribute being higher than the others, or on the existence of the entity as a whole with no single attribute dominating.

Consider the example illustrated in **Figure 5**. In this example there are two classification levels, CL1 and CL2, where CL2 requires more protection than CL1 (i.e., CL2> CL1). Let entities X and Y be classified per Case 2; X at CL1 and Y at CL2. Let entity Z be classified per Case 1, where its classification is based on the attributes Par and Child. As previously explained, since entity X is rated as CL1, its identifier attribute is rated at CL1. Since entity Y is rated at CL2, its identifier attribute is rated as CL2. When these identifier attributes are imported into entity Z, they must be treated as being at the level of the entity to which they reference; that is, attribute Z.Par is at CL1 while attribute Z.Child is at CL2. Since CL2 is greater than CL1 it is the dominant classification level and entity Z inherits the CL2 rating and is treated as such. This means that the identifier attribute of entity Z is also treated as CL2 (e.g., Z.ID=0x23ED7A43). This is important if it is imported into another entity. This propagation of classification level continues whenever an attribute is imported.

The fact that entity Z relates entities X and Y has no affect on the entities containing the imported attribute (e.g., X or Y). Thus, entity X remains at rating CL1. This is because entity X, by itself, provides no information or insight about entity Z. Therefore, entity X is created and maintained in the CL1 environment. Further, when it is duplicated to the CL2 environment, it may retain its CL1 rating. However, entities Y and Z must be created and maintained in the CL2 environment.

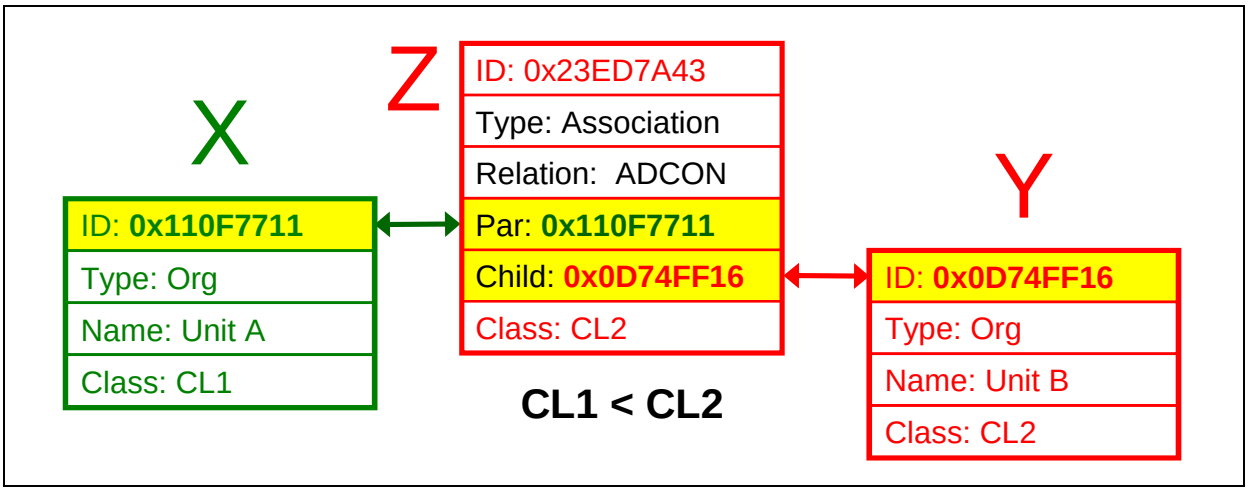

Figure 5: Classification Level Propagation Via Imported Attributes

This process may continue to other classification levels. For example, entities Y and Z may be duplicated to a CL3 environment (where CL3 > CL2) and maintain their CL2 rating. However, as expected, entities with a higher rating may never be moved to a lower environment, and this includes any of their attributes, to include identifiers. Even though an identifier may be unintelligent, it is still synonymous with the entity it identifies and it may be used to coalesce or aggregate other data that contains it. Therefore, the identifiers for entities Y and Z must always remain in the CL2 environment (or higher).

Figure 6 illustrates a variation of the example in **Figure 5**. In **Figure 6** entities X and Y are both rated at CL1 but the fact that they have an association (implemented via entity Z) is rated CL2. In terms of entity Z, this is an example of Case 2 (entity resolution) because the existence of the association is to be hidden. Notice that the only change to **Figure 6** is that attribute Y.Class=CL1. Observing only entity Z, there is no way to infer from its attributes whether it is rated CL2 due to Case 1 or Case 2. This may only be discerned when the imported entities are observed, and then one still can not know which case is in effect if one of the imported attributes is at the same level as the importing entity. This means that in **Figure 5**, entity Z could be rated CL2 regardless of the ratings of the imported attributes or because of the imported attribute. The only way to discern

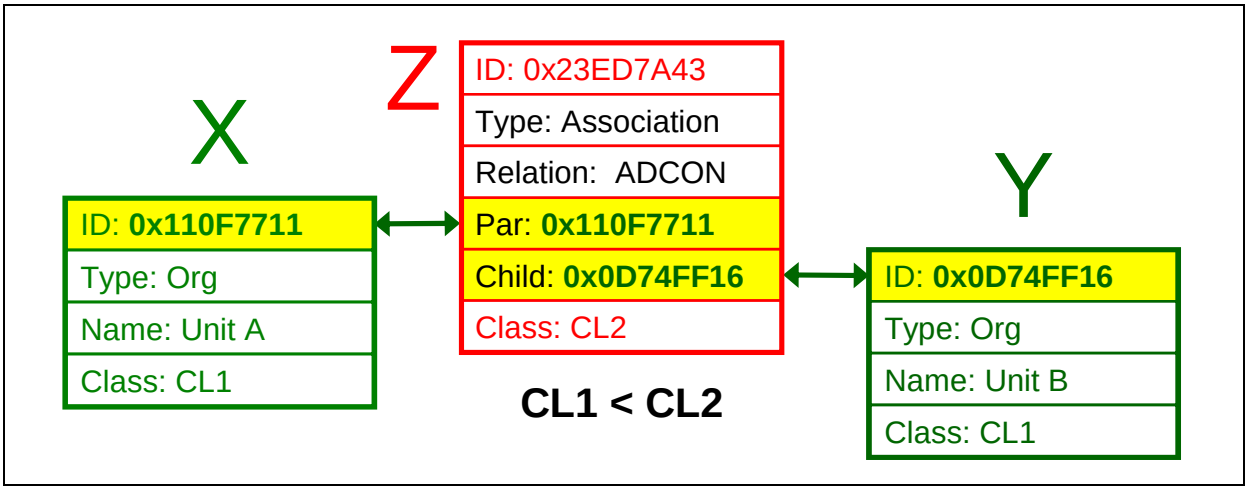

Figure 6: Classification Level Independent of Imported Attributes

this level of detail is if every attribute includes meta-data about its rating. But from an identifier perspective, it doesn't matter. In both cases the identifier attribute of entity Z is rated as CL2. In **Figure 6**, the entities X and Y are created and maintained in the CL1 environment and their identifier attributes are treated as CL1. However, entity Z must be created and maintained in the CL2 environment and its identifier attribute must also remain at the CL2 level (or higher). Further, any processing result (e.g., from a query) that includes the value of Z.ID must be rated at CL2 or higher. Fortunately, this behavior is logical and intuitive.

Figure 7 provides a multi-level example. A graph is presented that could represent an organization chart. As depicted in **Figure 4** through **Figure 6**, the boxes could denote organization entities (i.e., X and Y) and the lines association entities (i.e., Z). Every entity has an associated level, L1 - L3, with boxes having labels and lines having a unique type (dotted, solid, and dashed) corresponding to levels. The lines are also numbered. Levels have a precedence with L1 < L2 < L3 with Level 3 requiring the most protection. Quadrant I (upper left) shows all entities for all level.

In practice, Level 1 data would be created in that environment and passed on to the Level 2 environment. At the Level 2 environment, more data would be added and then the entire data set

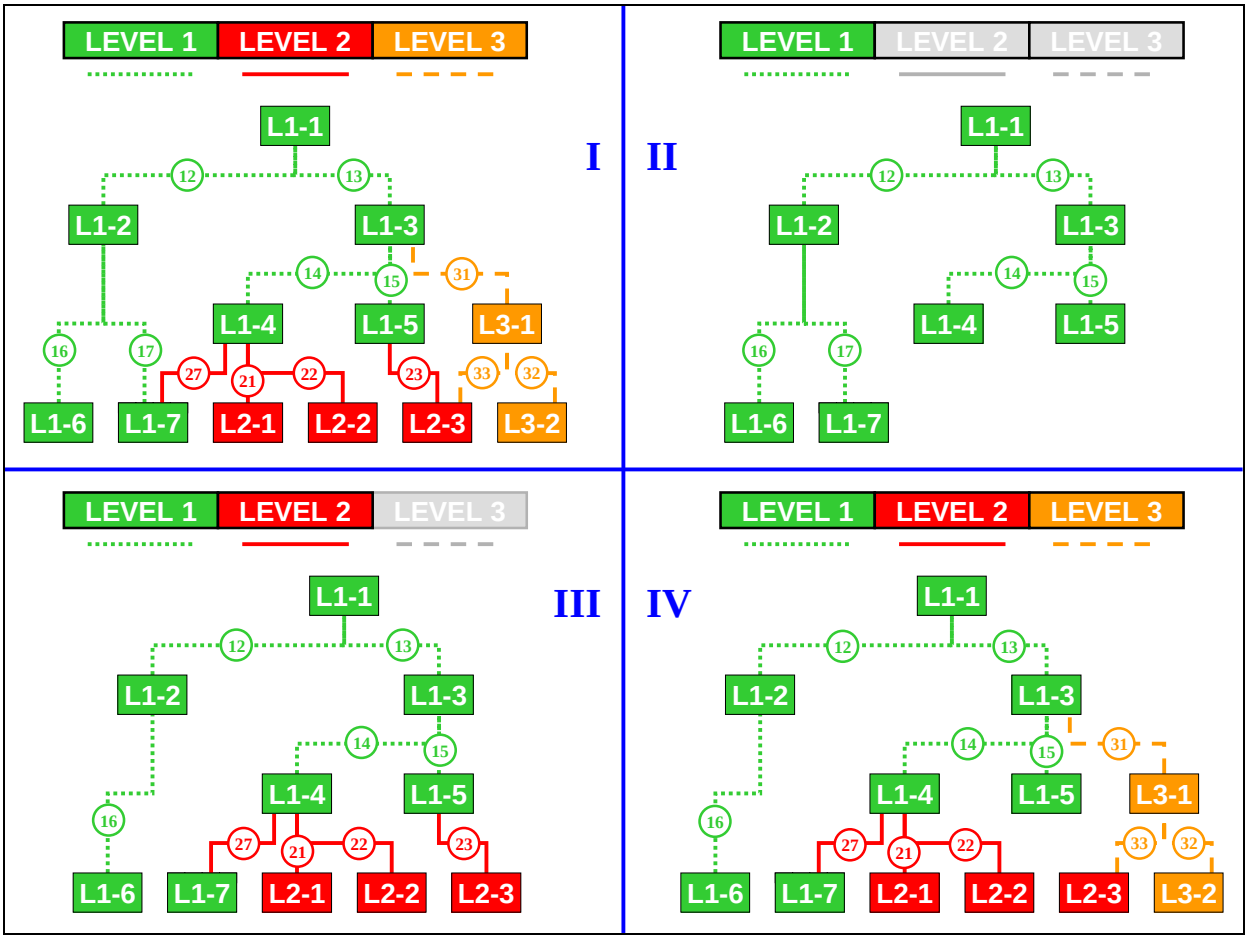

Figure 7: Examples of Multi-Level Data.

passed on the Level 3 environment. At the Level 3 environment, more data would be added and then the entire data set passed on to higher environments and this process can continue. Quadrant I illustrates the result of such a sequence of events with all the entities displayed.

Quadrant II (upper right) shows the data available at the Level 1 environment. At this point no Level 2 or 3 data is included in the data set. Therefore, this is a diagram of the "org chart" from a Level 1 perspective and depicts the data that would be duplicated to Level 2.

Quadrant III (lower left) presents the tree structure containing Level 1 and Level 2 data, but with Level 2 data prevailing when a conflict occurs. In this example, an organization can only have a single parent under a given set of circumstances. As seen in Quadrant I, with both L1 and L2 included, Org L1-7 has two parent Orgs: L1-2 via association 17 and L1-4 via association 27. However, since association 27 has a higher precedence level than association 17 (L2 versus L1), it dominates. What this illustrates is that from a L1 perspective (Quadrant II) the parent of org L1-7 is org L1-2, but from a L2 perspective (Quadrant III) its parent is L1-4. This is because association 27 provides additional, more protected information. Notice that Orgs L1-7 and L1-4 are rated as L1 but association L27 is rated as L2. This is perfectly allowable and is an example of a Case 2 (entity resolution) rating on the association that is completely independent from the orgs. In other words, the relationship is protected at a higher level than either of the orgs and the property of the relationship itself is what is sensitive, not the orgs that it relates. This means that any reference to the identifier of association 27 is also rated L2 and if it is included in another entity, then that entity must be protected at L2 or higher.

Quadrant IV (lower right) presents the Level 3 filtered view of Quadrant I. Again, the data from Quadrant III is duplicated to Level 3 and more data is added. In cases where there are more than one parent association, the highest precedence association dominates resulting in a tree structure (a single parent per organization). Therefore, Quadrant IV is a diagram of the "org chart" from a L3 perspective, versus L2 in Quadrant III, and L1 in Quadrant II. Notice the three different cases present in Quadrant IV. First, is the case in which all three entities (the parent and child organization and the association) have the same level, as in Orgs L1-1, L2-2, and association 12. Second, is the case in which the child org and the association have a higher level than the parent org, as in Orgs L2-1 and L1-4 and association 21. As previously explained, there is no way to distinguish from this information the reason for the level given to association 21. It could be because of the imported identifier from Org L2-1, or due to its own intrinsic sensitively (although it is immaterial in this case). Third, is the case in which the parent org and the association have a higher level than the child org, as in Orgs L3-1 and L2-3 and association 33. However, notice that there will never be a case in which an association will have a rating lower than either of its endpoints. This is because of the Case 2 (attribute resolution) property and the ratings of the imported attributes used to establish the association. An entity (e.g., an association) will always have a rating greater than or equal to the highest rating of its imported attributes, and this property is propagated to other entities that import identifiers with higher protection levels than their own. However, as illustrated by entity L1-3, having ones identifier imported does not change the protection level of the entity. It is the circumstances of the importing entity, not the imported entity, that determines the results of the interactions involved and the resulting overall protection level.

5. Summary

This paper describes a set of precepts used to define the process of handling data tagged with unique identifiers exchanged across security domain boundaries. Enterprise-wide identifiers are data identifiers whose scope of uniqueness extends over the complete enterprise. This wide scope allows one to quickly and easily isolate an entity across a vast infosphere because the identifier is synonymous with the entity. Just as important, enterprise-wide identifiers facilitate the gathering and integration of entities that use or refer to the entity via this identifier. Two general cases are defined to describe how a level of protection is ascribed to an entity. Case 1 is "attribute resolution" and refers to the situation in which one or more attributes of an entity can be cited as the cause for assigning a protection rating. Case 2 is "entity resolution" and refers to the situation in which the entity is protected as a whole and no particular attribute can be cited as the cause of the protection level. Often, Case 2 is the result of a desire to hide the existence of the entity. The basic rule stipulated is that an identifier, when serving as an attribute (not just a lone value), must be treated at the same level of protection as the entity it identifies. The leads to the propagation of protection levels as attributes from one entity are imported into other entities. An entity with an imported attribute must be protected at a level greater than or equal to the level of that imported attribute. This has no affect on the entity from which the identifier attribute was imported, but only on the importing entity. These simple rules define how a protection level is manipulated when integrating data and constrains how the integrated data, and particularly its identifier, must be treated.

6. References

DOD Directive 8320.03, *Unique Identification (UID) Standards for a Net-Centric Department of Defense*, 23 March 2007; USD(AT&L)/USD(P&R),
see: http://www.dtic.mil/whs/directives/corres/pdf/832003p.pdf.

DOD Instruction 8260.03, *Organizational and Force Structure Construct (OFSC) for Global Force Management (GFM)*, 23 August 2006, Under Secretary of Defense (Personnel and Readiness).
See: http://www.dtic.mil/whs/directives/corres/html/826003.htm.

GFM XML Schema Definition (XSD) v3.4.1.b, Posted 9 Feb 09.
Available via the Global Force Management Data Initiative Wiki page.
See: https://www.intelink.gov/inteldocs/browse.php?fFolderId=22991

Chamberlain, Sam; "An Enterprise Identifier Strategy for Global Naming Across Arbitrary C4I Systems" *Proceedings of the 6th International Command and Control Research and TechnologySymposium*; US Naval Academy, Annapolis, MD; 19-21 June 2001.
See: http://www.dodccrp.org/events/6th_ICCRTS/Tracks/Papers/Track2/059_tr2.pdf