CORRELATED DATA INFERENCE in ONTOLOGY GUIDED XML SECURITY ENGINE

Csilla Farkas **Andrei Stoica**

Information Security Laboratory
Department of Computer Science and Engineering
University of South Carolina, Columbia
{farkas, stoica}@cse.sc.edu[1]

Abstract

In this paper we examine undesired inference attacks from distributed public XML documents. An undesired inference is a chain of reasoning that leads to protected data of an organization using only publicly available information. We propose a framework, the **O**ntology guided **X**ML **Se**curity En**gin**e (Oxsegin), and algorithms to detect and prevent undesired inference attacks. Oxsegin uses the Correlated Inference Procedure to detect correlated information that may lead to undesired disclosure. The system operates on the DTD's of XML documents, and uses an ontological class-hierarchy to identify tags that may contribute to security violations. A security violation pointer is assigned to a set of tags that may contribute to a possible security violation. The likelihood of a detected security violation is measured by a confidence level coefficient attached to the security violation pointers.

Keywords: XML security, ontology based inference attack, data aggregation, correlated data inference, multi-level XML security

1. INTRODUCTION

Information systems have become a fundamental part of our everyday life. During the last few years the number of distributed applications using eXtensible Markup Language (XML) increased; the concept of Semantic Web emerged [16]. Ontologies [2, 17] support applications to access data without human assistance from several distributed sources and over different software platforms. The amount of data available over the Internet increases proportional with the demand for information. While individual data units are usually carefully analyzed not to disclose any confidential information, correlated data may allow unintended disclosure of confidential information.

[1] This work was partially supported by the National Science Foundation under Grant No. 0112874

XML security follows two main research trends: (*i*) Document Instance Security for digital signatures [14] and encryption [15] and (*ii*) Access Control Models for multi-level XML documents [9, 10, 11, 12, 13]. The main focus of these works is how to assign and enforce access permissions (e.g., security classification labels) to XML documents. Current techniques, however, do not consider the security implications of automated correlation of large amount of machine-understandable data.

To provide interoperation among large, distributed XML document repositories ontology-based query engines are being developed. The retrieval of the information is based on data semantics and requires minimum knowledge about document structure and syntax. Ontologies [1] include formal specification of concepts, definition of terms, relationships between data, vocabulary of concepts in a taxonomic structure, attributes of concepts, logical axioms, and other related information for a specific knowledge domain. They unify the different syntaxes and of the documents and supply background knowledge for query answering [3]. XML data retrieval is currently supported by several query languages, such as Lorel for XML [7], XML-QL, and XQL [8]. One of the research objectives for XML query engines is to use comprehensive ontologies to retrieve information based on the meaning of the query rather than the exact syntax [2, 5]. The query engines employ ontologies to derive additional knowledge using a deductive inference system.

Unfortunately, techniques, that support interoperation, may also lead to unintended and undesired inferences. Intuitively, an undesired inference occurs when a user is able to infer non-permitted information from intentionally disclosed data and available ontologies. This inference threat is similar to the inference problem in traditional databases, where the ontology corresponds to the external domain knowledge. However, due to (*i*) the dynamic nature of the Web, (*ii*) the large amount of information to be processed, and (*iii*) the fact that the owner of the sensitive information does not have control over all publicly available data that may lead to undesired inferences, traditional inference control techniques are insufficient to provide protection against undesired inferences. Up to date, small-scale data availability and the lack of automated data correlation tools limited the threat of unwanted inferences via external domain knowledge. The impact of automated XML document correlations from large distributed databases using ontologies has not been yet fully addressed from the information security point of view.

Our research targets the security impact of the ontology enhanced XML tools over large, distributed XML databases. We show that it is possible to use ontologies to mount specific data inference attacks on XML data. We develop techniques to detect and prevent attacks due to correlated data under different format. To prevent these attacks, we propose the **O**ntology guided **X**ML **Se**curity En**gin**e (Oxsegin). Oxsegin is a probabilistic engine that computes security violation pointers with associated confidence level coefficients. We develop and incorporate in Oxsegin algorithms and procedures to detect correlated data from a large collection of XML documents using the concept hierarchy from the ontology module. We also provide a framework to compute the associated confidence level of a security violation pointer based on correlated data, where the confidence level indicates the likelihood of the security breach.

The rest of the paper is organized as follows: Section 2 presents an example of ontology-guided attack using public domain data. Section 3 describes the architecture and functionality of Oxsegin. Section 4 gives the technical details for the correlated data inference process in the security engine. Finally we conclude and propose future research in Section 5.

2. ONTOLOGY-BASED ATTACKS IN XML DATABASES

Undesired inferences in multilevel secure databases have been studied extensively (see [18] for an overview). The inference problem is to detect and remove inference channels that lead to disclosure of unauthorized data by combining authorized data and meta-data. In traditional databases, the security officer has complete control over all organizational data, thus allowing the modification of their security classification of data or deny access to data if necessary to remove any unwanted inferences. In Web environment, where correlated data may come from several, independent sources, only a small portion of publicly available data is under the control of the owner of the sensitive information. Therefore, elimination of a detected inference channel may require other than information technology response to limit the possible damage. Nevertheless, the detection of a possible security breach via undesired inference is important.

We assume that organizational data contains both public (e.g., available from the Web) and confidential (e.g., available only to some of the users) data[2]. Before releasing the public data the organization wants to ensure that others will not be able to combine this public data with other publicly available data on the Web (e.g., other websites) to gain access to confidential data. If such disclosure is detected, appropriate response is performed. Response may range from declining the intended release of the public data or perform non-IT based countermeasures.

To perform an attack, attackers must acquire ontologies based on the type of sensitive information they target. Then, they employ a regular web crawler to browse public data and use the ontology to unify the information. Based on the available information and the correctness and details of the ontology it is possible that attackers successfully derive data that is not permitted for them.

Consider the document fragment (Figures 1.a) extracted from a database carrying information for upcoming air-shows. This document provides information, like the address and driving directions to military bases (Base_X) where an air-show is held. The second document fragment (Figure 1.b), extracted from a local State Division for Health Administration, shows a map of drinking water basins within a given state. Finally, the third fragment (Figure 1.c), is part of a sensitive document, containing data about the locations of the water sources for several military bases, including Base_X. The security requirement of the military is that the information about the water reservoirs of military bases should only be accessible by authorized users. The air-show information (fragment 1) is available on-line and the drinking water basins information (fragment 2) is outside of the military protection domain and publicly available. Indeed, our example is based on data available on existing Web site but we replaced the real data with fictional values.

[2] For simplicity, we only deal with public and confidential security labels that represent a total order. However, the presented techniques are applicable for the multilevel lattice-based models as well as to discretionary and role-based models.

A possible ontology for this attack unifies the <waterSource> with <basin>, <fort> with <base> and <address> with <district> tags. Using this correlation, the attackers gain access to secret information (association between the Base_X and its water source in Basin_Z), without any access to the critical infrastructure database. Going further, this type of information may be used in conjunction with a set of water chemical contaminants published by the Environmental Protection Agency along with possible commercial products that can supply these chemicals (details about this inference are purposely left out). Finally, the complexity of this attack is reduced by the simplicity of the ontology and uniform access to online resources.

Air-show information Figure 1.a	Drinking water basins Figure 1.b	Critical Infrastructure Figure 1.c
<?xml version="1.0"?> <show> …….. **P** <fort> Base_X </fort> <address> District_Y </address> </show>	<?xml version="1.0"?> <waterMap> …… **P** <district>District_Y </district> <basin>Basin_Z </basin> </waterMap>	<?xml version="1.0"?> <infrastructure> …….. **S** <base> Base_X </base> <waterSource> Basin_Z </waterSource> </infrastructure>

Figure 1.: Undesired Inference from Public Data

3. ONTOLOGY GUIDED XML SECURITY ENGINE

The motivation for the design of Oxsegin was to assist security officers and database administrators to securely update XML databases by identifying possible security violations from illegal inferences. Oxsegin uses a probabilistic inference engine with varying precision levels. Oxsegin indicates the possibility of unwanted inferences where the correlated data from the test files (publicly available data) matches the reference file (protected, confidential data). If unwanted inference is detected, some to the test files must not be released or non-IT response needs to be performed.

The security engine has four main components: the *Probabilistic Inference Module* – PIM, the *User Defined Inference Parameters Module* – UDIPM, the *Ontology Module* and the *XML Database Access Module*. The *Input and Feedback Module* - IFM is not incorporated in the Oxsegin architecture. The IFM functionality is to supply the reference and test XML structures, the set the inference parameters and decide the appropriate actions if a security violation is detected. Development of response policy to detected security violations is outside of the scope of this paper.

PIM computes possible security violation pointers between the reference document and the set of test documents. Intuitively, a security violation pointer indicates tags from the corresponding reference and test DTD files that might constitute unwanted inferences. For each security violation pointer, PIM computes an associated confidence level coefficient that reflects the likelihood of security violation involving the set of tags. UDIPM allows the security officer to define different inference processing parameters that will control the complexity of inference analysis. The inference uses the semantic formalism and concept hierarchy supplied by the Ontology module. The class hierarchy can be a general-purpose ontology or a custom build hierarchy to derive a specific attack.

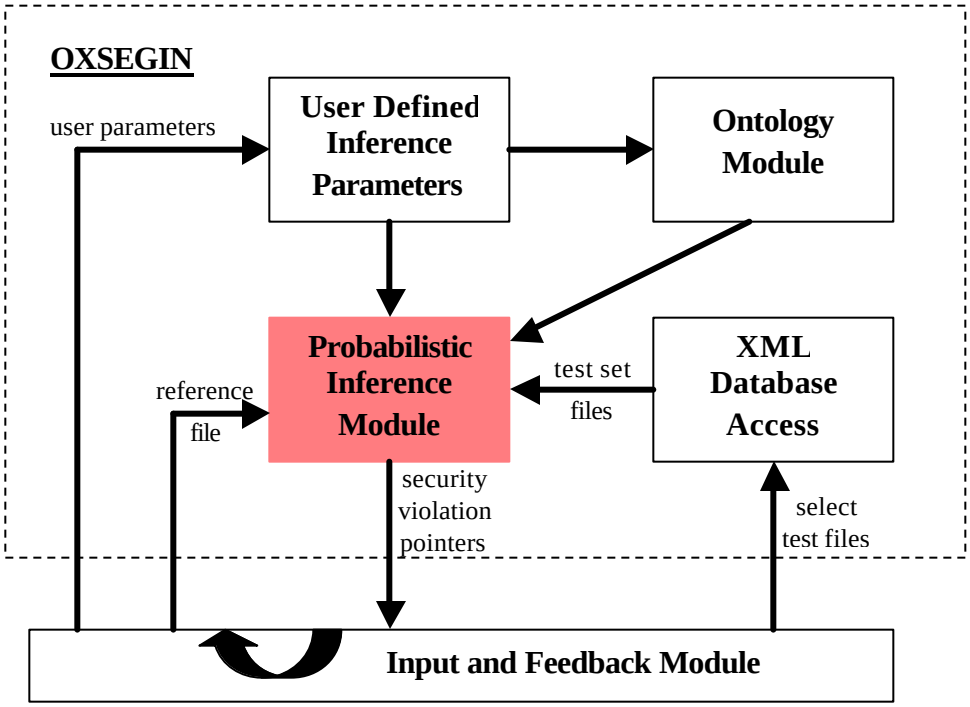

Figure 2: Oxsegin Architecture

The XML Database Access module represents a gateway to a collection of XML documents. There are no specific format or access requirements. The XML database can be the local, public document repository or files accessed via HTTP within a given web domain. As a result, Oxsegin can be used to securely publish documents over the web. The reference DTD structure corresponds to the protected document and the test DTD structures covers the set of all documents from the public domain, the security analysis determines the existence of undesired inferences.

3.2 Probabilistic Inference

PIM uses a set of procedures to identify security violations employing the ontology module to guide the inference process. Section 4 describes in full details the Correlate Inference Procedure. The Ontology module input is used to abstract the concepts represented by the tags within the DTD files. A security violation pointer SVP is assigned to every unwanted inference. The confidence level coefficient CLC is computed for each SVP, based on the set of probabilities corresponding to the concepts in the ontology, the relative position of the tags in the DTD files, and the relative position of the concepts in the ontology class hierarchy.

Definition 3.1 Security Violation Pointer
A Security Violation Pointer (SVP) is a set of tags $T = \{t_1 \ldots t_N\}$ that represent a possible security violation via unwanted inference.

Definition 3.2 Confidence Level Coefficient
The Confidence Level Coefficient (CLC) of an SVP is the likelihood of the inference involving the tags of the SVP.

Within a DTD, we distinguish between syntactically identical tags but at structurally different location. We define all tags as a pairs, containing the tag's name and the tag's path information from the root node of the DTD. For clarity, in the following we omit the path information unless it is needed to differentiate between the tags.

To formalize ontologies we adapt the use of Frame Logic [6] as the conceptual modeling language. It accommodates cardinality constraints for attributes and relationships in different granularities. We assume that the security officer assigns a weight to each concept in the ontology class hierarchy to differentiate between less and more specific concepts from the perspective of the protected sensitive information. The more specific a concept is, the larger the weight assigned to it. The root of the ontology class-hierarchy has a minimal weight since it is the least specific concept. Concepts that are relevant to the given knowledge domain and the specific security requirements usually carry larger weights. After the security officer assigns the weights for each concept, the system computes the set of the associated probabilities. Probabilities calculated for each concept reflect the likelihood of the same syntactic forms to represent the same semantic concepts, and are calculated by normalizing the weights assigned to each concept.

Definition 3.3 Ontological Abstraction Level
Given the concept C from ontology O, the Ontological Abstraction Level of C, denoted as OAL(C), is n if C is located at depth n in the corresponding ontology class hierarchy. The root concept C_R of the class-hierarchy has $OAL(C_R) = 0$.

Definition 3.4 Base Ontological Abstraction Level
The Base Ontological Abstraction Level of a tag t, denoted as BOAL(t), is the OAL of the concept C contained within the tag t.

Definition 3.5 Abstracting a concept N steps
A concept C from an ontology O is abstracted N steps when it is replaced N times by its immediate parents in the corresponding ontology class-hierarchy.

Definition 3.6 Container and Data Tags
A container tag is an XML tag that holds only structured information in the form of other XML tags and has no tags attributes. A data tags is an XML tag that contains at least one unit of information. A data tag may contain data and container tags.

4 CORRELATED INFERENCE PROCEDURE

In this section we propose an inference procedure that detects undesired inference attacks within a particular knowledge or semantic domain. The Correlated Inference Procedure detects ontology-based attacks similar to the one described in Section 2. The procedure checks a reference DTD structure (corresponding to the classified information) against a set of test DTD structures (corresponding to the publicly available information) by abstracting and unifying tags using the concepts knowledge supplied by the ontology.

The main data structure used by the Correlated Inference Procedure is the ***I**nference **A**ssociation **G**raphs* (IAG). Intuitively, IAG represents the associations among tags of an XML DTD structure. The nodes of an IAG correspond to the XML data tags and the edges represent associations between the tags. Figure 2. represents the IAG corresponding to the XML files in Figure 1.

Each association has an attached Association Probability Coefficient (APC) that reflects the likelihood the corresponding nodes represent related concepts. In addition, associations can be classified according to the security policy of the organization. A security violation pointer identifies associations of different IAGs where each association is among the same tags but has different security classification. Such associations represent cases where users can derive information in one set of documents while they are disallowed to access the same information in a different set of documents.

Definition 4.1 XML Association
Given a parent tag P with the security label L_P and any two of its children S_1 and S_2 in the XML DTD structure, P defines an XML association between S_1 and S_2. The association has a corresponding security label L_P and P represents the association source.

Definition 4.2 Association Probability Coefficient
The Association Probability Coefficient, denoted as APC, corresponding to an association between tags S_1 and S_2 with an association source P, represents the probability that P is used to semantically correlate tags S_1 and S_2.

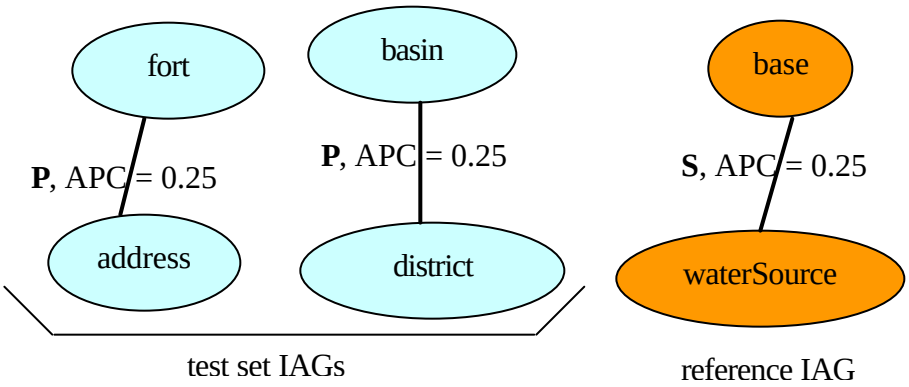

Figure 2: Inference Association Graphs IAGs

Definition 4.3 Inference Association Graph
The Inference Association Graph of an XML DTD structure, denoted by IAG=(V, E), is a graph with nodes V (data tags of the XML) and edges E (associations among the tags). Each edge is labeled with a pair (L_P, APC_P), representing the security label and probability coefficient of the association source tag.

Definition 4.4 Document Structure Level DSL(t)
Given a tag t from a DTD tree D, the document structure level of t in D, denoted as DSL(t), is the maximum depth of the sub-tree rooted at t. All the leaves l_1, l_2, …, l_k in the DTD have DSL(l_i) = 0.

Note, that it is always possible to find an XML association between any two tags in a DTD structure since the root tag is the parent for all tags in the DTD tree. However, this type of remote association is rarely relevant. In general, it is reasonable to assume that APCs decrease with the distance between the associated elements and the source. Algorithm 1. gives the formal description of the procedure to build the IAG. To reduce the complexity of the inference process, the algorithm limits the number of tags considered for XML associations. Associations are considered only if the relative difference between the tags and the association source in the DTD tree is less than MaxDepth. MaxDepth is a variable set by the security officer according to the specifics of the domain of the XML DTD structure.

```
Algorithm 1: Build IAG
Input: XML DTD structure
Output: IAG
BEGIN
   FOR all data tags Di DO
      Create a corresponding node Vi
   FOR all tags Ti DO
       FOR all Vj and Vk such that Dj and Dk successor of Ti and
       Depth(Dj)-Dept(Ti)<MaxDepth, Depth(Dk)-Dept(Ti)<MaxDepth DO
          Create the edge e between (Vj, Vk)
          Label e with (LTi, APCijk)
       END FOR
   END FOR
END
```

APC is calculated using the distance of the data tags from the association source, their relative distance and document structure level.

$$APC_{ijk} = \frac{1}{1+Depth(D_j)-Depth(T_i)} * \frac{1}{1+Depth(D_k)-Depth(T_i)} * \frac{1}{1+\left\|Depth(D_j)-Depth(D_k)\right\|} * \frac{1}{DSL(D_j)+1} * \frac{1}{DSL(D_k)+1}$$

The first two coefficients $\frac{1}{1+Depth(D_j)-Depth(T_i)}$ and $\frac{1}{1+Depth(D_k)-Depth(T_i)}$ in the definition of APC_{ijk} quantify the relative depth difference in the DTD tree between the associated tags and the source of the association. APC decreases with the distance between the tags and the association source. The third coefficient in the definition of APC_{ijk}, $\frac{1}{1+\|Depth(D_j)-Depth(D_k)\|}$, quantifies the relative depth difference between the associated tags. Tags at the same depth have a corresponding APC larger than tags at different depth in the DTD tree.

The next two coefficients $\frac{1}{DSL(D_j)+1}$ and $\frac{1}{DSL(D_k)+1}$ quantify the structural complexity of the associated tags. Tags that represent the root of larger sub-trees are more likely to be container tags, and this reduces the relevance of any association involving them.

```
Object[].                    OAL=0    WGT=1     OP=1/50
   waterSource :: Object     OAL=1    WGT=15    OP=15/50
      basin :: waterSource   OAL=2    WGT=1     OP=1/50
   place :: Object           OAL=1    WGT=15    OP=15/50
      district :: place      OAL=2    WGT=1     OP=1/50
      address :: place       OAL=2    WGT=1     OP=15/50
   base :: Object            OAL=1    WGT=15    OP=15/50
      fort :: base           OAL=2    WGT=1     OP=1/50
```

Figure 3: Ontology represented with Frame Logic statements

After building the IAG for each XML DTD structure in the test set, the ontology is used to integrate them into a single structure - the test set IAG. The Frame Logic statements in Figure 3. represent the ontology associated with the knowledge domain of the XML DTD structure in Figure 1. Each concept is shown with the associated ontology abstraction level OAL, weight WGT, and probability OP. If the XML DTD structures in the test set belong to the same knowledge domain, abstracting the tag names may create pairs of duplicated nodes among different IAGs. Eliminating the duplicated nodes collapses the test set IAGs into a single compact structure. Each node in the IAG has an attached Concept Abstraction Level coefficient (CAL). Intuitively, CAL reflects the likelihood that the new concept is an abstract representation of the tag that is replaced. For the initial concepts in the XML DTD structure, CAL = 1. Then for each abstraction, CAL is modified using the probability of the new concept in the ontology.

Definition 4.5 Concept Abstraction Level
The concept abstraction level (CAL) is the likelihood that the concept from the ontology hierarchy is an abstract representation of the initial XML tag name. For repeated replacements, CAL is the probability the present concept is an abstract representation of the original tag name.

Given the tree structure of the XML documents as well as the ontology hierarchy, all tags eventually collapse into a single node if abstracted to the root of the ontology. To prevent this from happening, the Correlated Inference Procedure has a set of restrictions on the abstraction process and the tags that it uses. The concepts are only abstracted within two predefined OAL limits (see definition 3.3): MaxOAL and MinOAL. MaxOAL is usually set to the depth of the ontology hierarchy tree while the MinOAL is set according to the specifics of the ontology. Usually, MinOAL is the average ontology depth of the concepts targeted by the inference attacks and is set by the security officer based on a particular knowledge domain. The second restriction on the abstraction process is based on the targeted tags. Tags located towards the root of the XML document are usually container tags, mostly used for structuring the document and rarely involved in semantic correlations. Since this cannot be made a general rule because is highly dependent on the XML document, again the security officer assigns a maximum level within the XML structure to consider tags in the abstraction process – MaxDSL where DSL denotes the document structure level.

Integrating the test set IAGs simulates the natural human brain inference process in three distinct stages. In the first stage the concepts associated with XML tags are abstracted, unifying same notions originally under different syntactic forms. In the second stage, by eliminating the duplicated nodes and collapsing the multi-structure IAGs, the system simulates the inference link between multiple files with related data. In the third stage the system performs a transitive correlation to simulate linking XML tags through similar abstract concepts. This is the most significant step since the security violation pointers are based on edges created by the transitive correlation. The transitive correlation relates two tags through an XML association (IAG edge) with a common third tag. Since the targeted inference is usually between multiple XML DTD structures, it follows naturally to perform the transitive correlation after duplicated node reduction. Algorithm 2. gives the formal description of the Correlated Inference Procedure.

Each edge added in the transitive correlation of the test set IAG represents a possible illegal inference. The Correlated Inference Procedure checks all these edges against the reference IAG to identify security violation pointers. The test for security violation pointes is performed on edges, since the edges represent valid XML associations. Each edge added to the test set IAG by the transitive correlation is compared to all edges in the reference IAG. The system places a security violation pointer (SVP) on pairs of edges between similar nodes if the reference edge security label dominates the test set edge security label. Intuitively this means that an association from the reference DTD structure is classified at a higher security level than an association among the test DTD structures discovered by the transitive correlation procedure. The edges are matched for a security violation employing again the ontology hierarchy to abstract concepts for each tested edge. The tags are abstracted up to a minimum ontology abstraction level MinOAL to avoid matching all concepts at the ontology root. This step is not computational demanding considering the relative limited number of concepts in the ontology relative to the number of tags in the test DTD structures set. Each SVP has a confidence level coefficient CLC computed based on the APC of the edge and the CAL of the nodes.

The last coefficient in computing CLC, $\left(1-\left\|CAL_{max} - CAL_{min}\right\|\right)$ quantifies the relative difference between the maximum and minimum level of abstraction for the concepts in the XML associations. Concepts on the same level of abstraction in the ontology hierarchy have a higher associated CLC.

```
Algorithm 2: Correlated Inference Procedure
Input: Test and Reference DTD structures IAGs
Output: Security Violation Pointers
BEGIN
 FOR ALL tags Ti DO CALi = 1
 FOR X=MaxOAL DOWNTO MinOAL DO
   FOR ALL tags Ti such that DSL(Ti)<MaxDSL, OAL(Ti)=X DO
     Abstract Ti by 1, CALi = CALi* OP(Ti concept)
     IF duplicated nodes THEN
       Eliminate duplicated nodes
         FOR ALL tags Tj such that Tj=Ti DO
           Remove Tj and direct all edges to Ti
           CALTi = min[CALTi,CALTj]
         END FOR
       Transitive correlation
         FOR ALL tags Tj and Tk where Tj,Tk connected with Ti DO
           Let  L = max[LTi-Tj,LTi-Tk] and APC = APCTi-Tj * APCTi-Tk
           Connect Tj and Tk by en with label (L, APC)
           FOR ALL edges em in the reference set DO
             FOR ALL concepts ∈[MinOAL,MaxOAL] of em, en nodes DO
              IF em ≡ en THEN
                IF (security label en)<(security label em) THEN
                CALavrg = average CAL for en and em nodes
                CALmax = max CAL for en and em nodes
                CALmin = min CAL for en and em nodes
                CLC = CALavrg * APCen * APCem * (1 - ||CALmax - CALmin||)
                Place SVP on tags corresponding to nodes in the
                edges en, em with an associated CLC
         END IF
     END IF
   END FOR
 FOR ∀ SVPi such that CLCi > DSTcoef DO
   Perform data search on associated tags
   IF data match THEN
     CLCi = 1
 END FOR
END
```

Figure 4 shows the reference IAG and the integrated test set IAG corresponding to the IAGs in Figure 2 and the XML files in Figure 1. The tag <fort> was abstracted to <base> and the tag <basin> was abstracted to <waterSource>. Both tags <address> and <district> were abstracted to <base> inducing a transitive correlation between <base> and <waterSource>. The new XML transitive association between <base> and <waterSource> is classified public according to the Correlated Inference Procedure algorithm. This triggers a security violation between the test set and the reference IAG where the same association is classified secret.

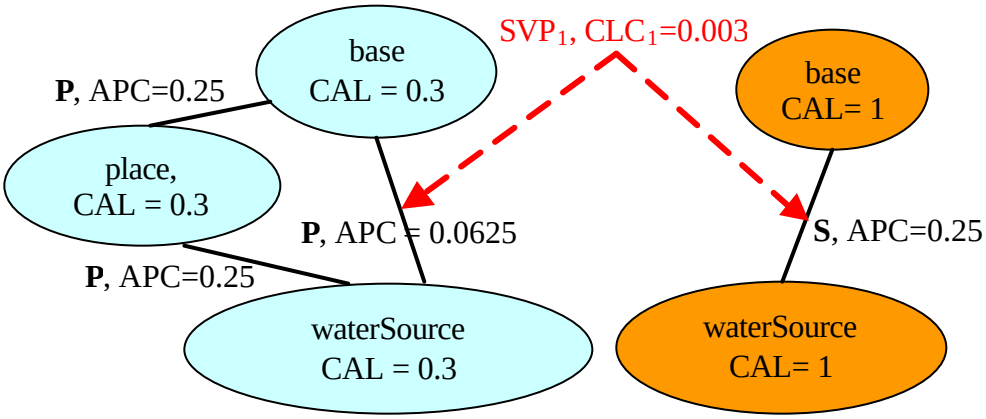

Figure 4: Unified Inference Association Graph

If the CLC corresponding to a particular SVP is above the Data Search Threshold coefficient (DSTcoef), the system provides low-level data granularity search. If data items associated with the reference and test set XML DTD structures match, the associated CLC is set to 1, the maximum confidence level. The low-level data search provides maximum security but also maximum processing complexity. High-level detection may produce false positive security violation pointers with high confidence coefficients. Data granularity search decreases the amount of false positives but does not guaranty to eliminate all of them. The Correlated Inference Procedure runs the analysis for security violation pointers on the DTD structure level. This represents an advantage for large XML documents databases where usually more than one document corresponds to any given DTD file. Operating at the DTD level is similar to high-level security detection with reasonable accuracy under reasonable computational complexity. For more accurate detection the procedure uses specialized data granularity search to identify security violations with maximum confidence level.

5. CONCLUSIONS AND FUTURE WORK

This paper presents a new method to prevent inference attacks in large XML databases. We show how ontologies can be used to implement automated attacks on large XML databases and develop methods and techniques to detect and prevent such attacks. Although ontological inferences have been studied from the perspective of providing interoperation, the security impacts of these new technologies have not been investigated and there are no tools to prevent these threats.

To the authors' best knowledge, Oxsegin is the first proposal to provide a semantically enhanced XML security framework. This paper adds a new component to the security engine to prevent inference attacks based on correlated data. The Correlated Inference Procedure runs a probabilistic algorithm to computes security violation pointers and their associated confidence level probability. The procedure can be tuned to run at different complexity levels to enhance the efficiency of the model.

In future work, we are developing a simulation of Oxsegin. We are planning to test the performance of the simulation against human analysis using both naïve users and domain experts. We expect our model to have accuracy similar to the accuracy of a domain expert. The main contribution of our model is to be able to handle large amount of semi-structured data that is infeasible by using human experts only.

References

[1] T.R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*. Vol.6, no.2, 1993. pp199-221

[2] M. Erdman and R. Studer. How to Structure and Access XML Documents with Ontologies. *Data and Knowledge Engineering, Special Issue on Intelligent Information Integration* (to appear)

[3] M. Erdman and R. Studer. Ontologies as Conceptual Model for XML Documents. *Proc. of the 12-th Workshop for Knowledge, Acquisition, Modeling and Management*, Banff, Canada, October 1999

[4] M. Erdmann, S.Decker. Ontology-aware XML Queries. http://www.aifb.uni-karlsruhe.de/~mer/Pubs/semantic-xql.webdb00.pdf

[5] B. Amann, I. Fundulaki, and M.Scholl, et at. Mapping XML Fragments to Community Web Ontologies. *Proceedings Fourth International Workshop on the Web and Databases* (WebDB'2001)

[6] M. Kifer, Georg Lausen, James Wu. Logical Foundations of Object Oriented and Frame Based Languages, *Journal of ACM* 1995, vol. 42, p. 741-843

[7] S. Abiteboul, D. Quass, J. McHugh, J.Widom, and J. Wiener. The Lorel query language for semi-structured data, *Journal of Digital Libraries*. Volume 1, 1997

[8] J. Robie, J. Lapp, and D. Schach. XML Query Language (XQL), *Proceedings of the W3C Query Language Workshop* (QL-98), Boston, 1998

[9] E. Bertino, S. Castano, E. Ferrari, M.Mesiti. Specifying and Enforcing Access Control Policies for XML Document Sources, *WWW Journal, Baltzer Science Publishers*, Vol.3, N.3, 2000

[10] A. Gabillon and E. Bruno. Regulating Access to XML Documents, *In Proc. IFIP WG11.3 Working Conference on Database Security*, 2001

[11] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. XML Access Control Systems: A Component-Based Approach, *In Proc. IFIP WG11.3 Working Conference on Database Security*, The Netherlands, August 21-23, 2000

[12] F. Dridi and G. Neumann. Towards access control for logical document structure. *In Proc. of the Ninth International Workshop of Database and Expert Systems Applications*, pages 322--327, Vienna, Austria, August 1998

[13] M. Kudo and S. Hada. XML Document Security based on Provisional Authorizations, *In Proc. of the 7th ACM conference on Computer and Communications Security*, Athens Greece, November, 2000

[14] P. Devanbu, M. Gertz et al. Flexible authentication of XML documents, *ACM Conference on Computer and Communications Security*, 2001

[15] XML Encryption Requirements, *W3C Working Draft*, 18 October 2001, http://www.w3.org/TR/2001/WD-xml-encryption-req-20011018

[16] T. B.-Lee, J. Hendler and O. Lassila, "The Semantic Web", Scientific American, May 2001

[17] "Ontology Inference Layer". http://www.ontoknowledge.org/oil/

[18] S. Jajodia and C. Meadows. Inference problems in multilevel secure database management systems. In Information Security: An integrated collection of essays, pages 570-584, IEEE Computer Society Press, Los Alamitos, C.A., 1995

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF)