PAMUKKALE ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ YIL : 2004
PAMUKKALE UNIVERSITY ENGINEERING COLLEGE CİLT : 10
MÜHENDİSLİK BİLİMLERİ DERGİSİ SAYI : Özel Sayı
JOURNAL OF ENGINEERING SCIENCES SAYFA : 57-61

VERİTABANI GÜVENLİK POLİTİKASI METAVERİ MODELİ OLUŞTURULMASI VE BİR UYGULAMA

Dilek Tapucu ARPAÇAY* Murat Osman ÜNALIR**
*İzmir Yüksek Teknoloji Enstitüsü, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 35437/Urla-İzmir
**Ege Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 35100/İzmir

ÖZET

Veritabanları üzerinde saklanan veriler, ilgili bilgisayar sistemi için tanımlanmış güvenlik politikaları ile korunmalıdır. Güvenlik politikaları bilginin değerini ve ilişkili riskleri karşılayacak şekilde ortaya konmalıdır. Bu çalışmada veritabanları için ortaya konulması düşünülen iki ayrı güvenlik politikası "Genel Olarak Kabul Edilen Sistem Güvenlik Prensipleri" ile açıklanmaktadır. Anılan politikalar için metaveriler ile metaveriler oluşturulmakta ve ilgili metaveriler, XML-Şema kullanılarak uygulamaya geçirilmektedir. Böylece her bir modele ait elemanların yapıları ortaya konulmaktadır. Gerçekleştirilen uygulama ile, organizasyonlar içinde belirlenen politikaların, değişen durumlar karşısında yeniden düzenlenmesinin kolaylaşacaktır. Bu sayede, metamodeller üzerinde yapılacak değişiklikler ile, ilgili politikaların süreç içinde devamlılığı sağlanabilecek ve metamodeller farklı sistemler arasında taşınabilecektir.

Anahtar Kelimeler : Veritabanı Güvenliği, Metaveri, Metamodel

DEVELOPMENT OF A DATABASE SECURITY POLICY METADATA MODEL AND ITS APPLICATION

ABSTRACT

Data stored in the database must be protected by pre-defined security policies designed for systems. The security policies must be described with the worth of knowledge and related risks. In this paper, two different security policies are explained by "Generally Accepted System Security Principles". For these policies metadata and metamodels are created and related metamodels are applied by using XML Schema. Then, the structures of elements for each model are exposed. For the realized application, determination of the policy in the organization, for changing conditions rearrangement will be easy. Thus, with the changes made in the metamodels, related policies will be continues in the process and metamodels are moved between different systems.

Key Words : Database Security, Metadata, Metamodel

1. GİRİŞ

Veritabanları toplanmış verilerin anlamlı bir şekilde düzenlenmesi ve örgütlenmesi ile oluşturulan sistemlerdir. Bunun için bilgi nesnelerinin yada sistem üzerindeki varlıkların tanımlanması ve birbirleriyle olan ilişkilerinin gösterilmesi gerekmektedir. Her veritabanı ancak kendisini kullanacak ilgili bilgisayar sistemindeki tanımlanmış, ve yetkilendirilmiş kullanıcılar için erişilebiliyor olmalıdır. Veritabanları üzerinde saklanan veriler, ilgili bilgisayar sistemi için önceden tanımlanmış güvenlik politikaları ile korunmalıdır. Güvenlik politikaları bilginin değerini ve ilişkili riskleri karşılayacak şekilde etkili bir güvenlik oluşturabilmek için ortaya konmaktadır (Al-Qahtani, 2000). Güvenlik politikaları, bilgi değerlerinin sahibinin görev tanımını yapmalı ve bu değerlerin güvenilirliğinin, kullanılabilirliğinin, bütünlüğünün bilginin sahibi ve diğer birimler için

önemini kapsamalıdır. Veritabanı güvenliği için ortaya konabilecek güvenlik politikaları başta verinin korunması için kullanıcı erişim haklarını sınırlandırmalı, ilgili bilgisayar sistemi için tanımlı kullanıcıların belirlenmiş rollere göre yetkilendirilmelerini sağlamalıdır. Bu çalışmada veritabanları için ortaya konulması düşünülen iki ayrı güvenlik politikası "Genel Olarak Kabul Edilen Sistem Güvenlik Prensipleri" ile açıklanacaktır. Anılan politikalar için "Metaveriler" oluşturulacak ve bu modellerin uygulama örneği XML Şema yapısı ile yapılandırılacaktır. XML genişletilebilir işaretleme dili olarak açıklanmaktadır ve meta taglardan oluşmaktadır. Meta tag yaklaşımı, XML temelli bilgiye XML Şema yapısı kullanarak standart kazandırmaktadır.

Anılan çalışma, metaveri metaveri ilişkisini veritabanı uygulamaları için zorunlu olan güvenlik kavramı ile açıklamaktadır. Hedef, tanımlı güvenlik politikalarını metaveriler aracılığıyla oluşturulan metaveriler ile ifade edebilmek, değişen durumlar karşısında daha önceden oluşturulan politikaların genişletilebilir, taşınabilir olmasını sağlamaktır. Böylece bir organizasyon için belirlenen güvenlik politikaları değişen durumlar karşısında yeniden yaratılmayacak, metaveriler üzerinde yapılacak değişiklikler ile ilgili politikaların süreç içinde devamlı olması sağlanacaktır.

2. VERİTABANI GÜVENLİĞİ İÇİN POLİTİKA OLUŞTURMA

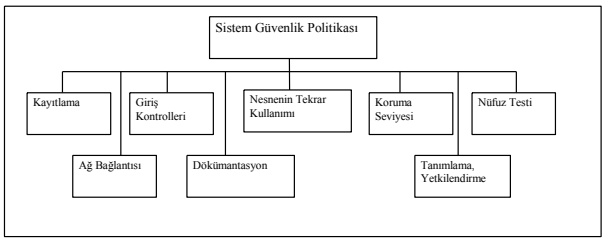
Bu çalışmada, veritabanı güvenlik politikası geliştirmede Genel Olarak Kabul Edilen Sistem Güvenlik Prensipleri (Generally Accepted System Security Principles "GASSP") GASSP'tan yararlanılmaktadır, GASSP prensipleri, politika geliştirme yaklaşımını ve bu konuda dikkat edilmesi gereken hususları ortaya koymaktadır (Doruk, 2001).

GASSP, Uluslararası Bilgi Güvenliği Fonu (I^2SF) tarafından desteklenmekte olan değişik ülkelerden bilgi güvenliği çalışanları, uzmanları ve bilgi sahipleri tarafından meydana getirilen bir komitenin çalışmaları sonucunda ortaya konmuştur. Bu prensiplerde hedef; günümüzün en değerli kaynağı olan bilginin güvenliğinin sağlanmasıdır. Anılan prensipler; kapsamlı, genel fonksiyonel ve detaylandırılmış prensipler olarak sınıflandırılmıştır. Kapsamlı prensipler bilgi güvenliğinin kavramsal amacını temsil eder. Sayıca azdır, temel türlerdir ve çok ender değişirler. Bilginin gizliliğini, doğruluk ve bütünlüğünü ve her zaman kullanılabilir olmasını sağlamak amacı ile geliştirilmişlerdir. Genel fonksiyonel prensipler bir veya daha fazla kapsamlı prensibin birleşmesinden oluşur ve sayıca daha fazladır. Özeldirler ve daha detaylı prensiplerin gelişimine öncülük ederler. Sadece teknolojideki ve etken konulardaki ana gelişmeleri yansıtarak değişirler. Detaylandırılmış prensipler ise bir veya daha fazla genel fonksiyonel prensibin birleşiminden oluşur. Sayıca çok fazladır, spesifiktirler ve gelişen teknoloji ile beraber sık sık değişirler (Doruk, 2001).

Genel fonksiyonel prensipler, bilgi güvenliğinin kavramsal amaçlarını temsil eden kapsamlı prensiplerden çıkarılmışlardır. Veritabanı güvenliği için bu çalışmada ortaya konan politikalar Genel fonksiyonel prensipler ile desteklenmektedir. "Erişim Kontrolü" prensibi ile, kullanıcı erişim haklarını sınırlandırmak için ortaya konulacak politika, "Sistem Bütünlüğü" prensibi ile, ilgili bilgisayar sistemi için tanımlı kullanıcıların belirlenmiş rollere göre yetkilendirme yapılmasını gösteren politika açıklanacaktır (Doruk, 2001).

3. OLUŞTURULAN GÜVENLİK POLİTİKALARI İÇİN METAMODELLER

Metaveri, "veri ile ilgili veri" anlamına gelmektedir (Tannenbaum, 2001). Böylece önce veri hakkındaki bilgiye sonra verinin kendisine ulaşılmaktadır. Bir bilgisayar sistemindeki alt yapıyı kontrol edebilmek için metaveri kavramı önem kazanmaktadır. Metaveriler ile metaveriler oluşturulmaktadır. Buna göre, metaveriler metaverinin arkasında yer alan ayrıntıları içermektedir. Metaverileri yaratabilmek için gereksinimler bilinmeli, tanımlanan nesneler arasındaki metaveri akışı gösterilmelidir. Oluşturulan modelin uzun süre yaşayabilmesi için metaverilerin genişletilebilir özelliğe sahip olması gerekmektedir.

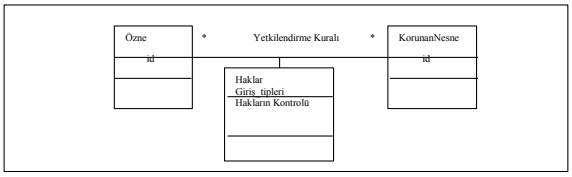


Şekil 1. Sistem güvenlik politikasının ayrıştırılması (Henning, 1996)

Bir bilgisayar sistemi için ortaya konan güvenlik politikalarına ilişkin metaverilerin oluşturulabilmesi için öncelikle metaverilerin tanımlanması ve bu tanımlı verilerin kontrol altında tutulması

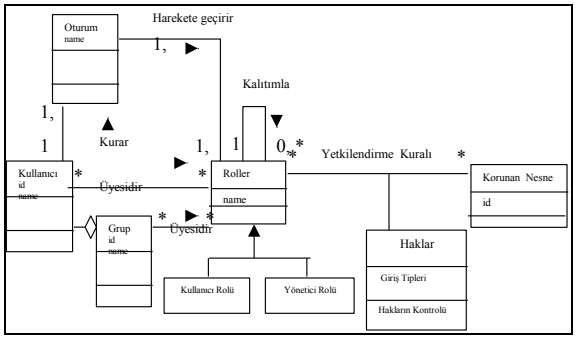
gerekmektedir. Metaverinin kontrol altında tutulması verinin hangi koşulda değişeceğinin, ve veri kalitesinin belirlenmesi ile sağlanır. Veri kalitesi "verinin yönetilebiliyor olması" anlamına gelmektedir (Henning, 1996). Veritabanları için veri yönetimi olduça önemli bir kavramdır, güvenliğin sağlanabilmesi için anılan veritabanına kimin, neden giriş yaptığı bilinmelidir.

Şekil 1'de bir bilgisayar sisteminde güvenliğin sağlanabilmesi için oluşturulacak güvenlik politikasının elemanlarının neler olduğu gösterilmektedir. Buna göre bir veritabanında giriş kontrollerine yönelik yapılandırılacak politika "kullanıcı erişim hakları sınırlandırılmalıdır" şeklinde olmalıdır. Böylelikle "erişim haklarının parola kullanımı ile sınırlandırıldığı", "sadece yetkilendirilmiş operatorler tarafından X, Y, Z operasyonlarının gerçekleştirileceği" anlaşılmaktadır. İlgili politika için oluşturulan metaveri Şekil 2'de gösterilmektedir (Euardo, 2001).



Şekil 2. Yetkilendirme şablonu (Euardo, 2001)

Veritabanları için ortaya konan ikinci politika tanımlama ve yetkilendirmeye yönelik "belirlenmiş rollere göre sistem kullanıcılarına yetkilendirme yapılmalıdır" şeklinde olmalıdır. Böylece "sistem üzerinde, sadece yetkili kullanıcıların ve sistem operatörünün kendilerine tanımlanan roller ile hareket edebilecekleri" ve "kullanıcıların kendileri için belirlenmiş roller dışına çıkamayacakları" anlaşılacaktır.



Şekil 3. Rollere göre giriş kontrol şablonu[5]

Anılan politikanın metaverii yine şekilsel olarak Şekil 3'te ifade edilmektedir. Böylece farklı iki güvenlik politikası için oluşturulan metaverisi ile XML-Şema dökümanlarının yaratılması olanaklı hale getirilmektedir.

4. XML-ŞEMA KAVRAMI

XML, World Wide Web Konsosiyum (W3C) tarafından yaratılmış, genişletilebilir işaretleme dili olarak açıklanmaktadır. Açık ve platform bağımsız, tag yapısı ile kolay okunur ve anlaşılır bir standarttır (W3C Recommendation, 2000).

XML-Şema dökümanı, kendisini yaratan XML instance dökümanı verilerinin tanımlandığı yer olarak bilinmektedir. Bir diğer ifade ile XML-Şema dökümanları, metaveri içinde yapısal tanımlamaların yapıldığı dökümanlardır.

Yapısal tanımlamalar ile,

- Bir XML dökümanında yer alabilecek elementler
- Bir XML dökümanında yer alabilecek attribute'lar
- Hangi elementlerin child elementler olduğu
- Child elementlerin kullanım sırası
- Child elementlerin sayısı
- Bir elementin empty olup olmadığı, metin (text) içerip içeremeyeceği
- Attribute'ların varsayılan değerlerini tanımları belirtilmektedir (W3C Recommendation, 2001).
- Yapısallığın sağlanması ile XML dökümanları daha güçlenmektedir.

XML Şema Tanımla Dili (XSD), XML dökümanlarının yapısı ve veri tipi tanımlanmasında kullanılır. XML Şema da veri tipleri ve elementlerin tanımı W3C XML Şemalarında Veri Tipi Tanımlama Kurallarına (W3C Recommendation, 2001) ve XML Şema Tanımlama Dili 'nin Şema Tanımlama Dili Kurallarına (W3C Recommendation, 2001) uygun olması gerekir.

5. META–MODELLERİN XML ŞEMA İLE İLİŞKİLENDİRİLMESİ

Veritabanı yönetim sistemleri dosyalar içinde verileri saklarlar. Her dosya kayıtlardan, her kayıt ilgili bir grup veriden oluşmaktadır. Her kaydın içerdiği alanlar ilgili veritabanının elemanlarını göstermektedir ve veritabanlarında alanlar arasında bulunan ilişkileri ve bu alanlar arasında bulunan ilişkileri göstermektedir. Tüm bu yapı

veritabanlarının mantıksal yapısı olan şema olarak adlandırılmaktadır (Al-Qahtani, 2000). Bir veritabanı XML Şemayı diğer sistemlere bilgi aktarmak amacıyla kullanabilir.

Bu çalşmada, veritabanları için oluşturulan güvenlik politikaları için metaverilerin uygulamaya dökülmesi XML-Şema dökümanlarının yaratılması ile sağlanmaktadır. XML-Şema dökümanları ile ortaya konan yapısallık insanlar ve bilgisayar arası iletişimde standart yapının oluşmasını sağlamakta, genişletilebilir alt yapının oluşmasını olanaklı hale getirmektedir. Bu çalşma için oluşturulan; "Yetkilendirme_Sablonu.xsd" , "RollereGöreGirişKontrolŞablonu.xsd " dökümanları, ilgili özellikleri ile birlikte aşağıda açıklanmaktadır.

Bir XML dökümanının ilk bölümünü "namespace" yapısı oluşturmaktadır. XML namespace, URI referansları ile tanımlanmış bir dizi isim topluluğu olarak açıklanmaktadır (Berners, 1998). Bu yapı, ilgili XML dökümanı içinde URI referansları ile belirtilen eleman (element) tipleri ve özellik (attribute) isimlerini sınırlamada basit bir metod sağlar. Her XML Şema dökümanındaki kök element bir elementtir. Bu şema elementi bazı nitelikler içerir.XSD dosyalarının başında hangi isim uzayını kullanacağı belirtilir. Aşağıda "Yetkilendirme_Sablonu.xsd" dökümanına ilişkin namespace yapısı yer almaktadır.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.Yetkilendirme_Sablonu.org"
xmlns="http://www. Yetkilendirme_Sablonu.org"
elementFormDefault="qualified">
```

Yetkilendirme Şablonuna ilişkin XML-Şema dökümanı Şekil-2 de gösterilen "Özne", "Korunan_Nesne" ve "Haklar" sınıfına ilişkin elemanları içermektedir. Aşağıda bu elemanların bir xsd dökümanı içinde nasıl yer aldığı belirtilmektedir.

```
<xsd:include schemaLocation="Ozne.xsd"/>
<xsd:include schemaLocation="Korunan_Nesne.xsd"/>
<xsd:include schemaLocation="Haklar.xsd"/>
```

Anılan XML-Şema dökümanınını tamamlanabilmesi için yukarıda belirtilen her eleman için tanımlayıcı özellikler arasında yer alan elemanın isimlerinin ve özelliklerin tipini "CDATA, ID, IDREF, ENTITY, NMTOKEN" gibi ortaya konmalıdır. Anılan kısıma ilişkin alınan örnekte "Özne" elemanı için id, isim, soyad, departman_adı gibi ilgili tagler ile ifade edilmekte ve aşağıda gösterilmektedir.

```
<xsd:element name="Ozne">
  xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Id"/>
        <xsd:element ref="Isim"/>
          <xsd:element ref="Soyad"/>
          <xsd:element ref="Departman_adı"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
<xsd:element name="Ozne">
 <xsd:sequence">
  xsd:element name="Id" type="xsd:integer"/>
  <xsd:element name="Isim" type="xsd:string"/>
    <xsd:element name="Soyad" type="xsd:string"/>
     <xsd:element name="Departman_adı" type="xsd:string"/>
 </xsd:sequence">
</xsd:element name="Ozne">
```

Rollere Göre Giriş Kontrol Şablonu için yapılandırılan ikinci XML-Şema dökümanı, Şekil 3'te gösterilen şablona ilişkin tanımlanan sınıfları ve bu sınıflar için belirlenmiş elemanları içerecek şekilde yapılandırılmaktadır. İlgili dokümanda "Kullanıcı", belirli kullanıcıların oluşturduğu "Grup", kullanıcıların, grupların ve oturumların sahip oldukları "Roller", bu rollerle ilişkilendirilen "Haklar" ve "Korunan_Nene"lere ilişkin eleman tanımlamaları yapılmaktadır. Aşağıda, "RollereGöreGirişKontrolŞablonu.xsd" dökümanında yer alan ilgili bölüm gösterilmektedir.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.
RollereGöreGirişKontrolŞablonu.org"
     xmlns="http://www. RollereGöreGirişKontrolŞablonu.org"
     elementFormDefault="qualified">

<xsd:include schemaLocation="Oturum.xsd"/>
<xsd:include schemaLocation="Korunan_Nesne.xsd"/>
<xsd:include schemaLocation="Haklar.xsd"/>
<xsd:include schemaLocation="Roller.xsd"/>
<xsd:include schemaLocation="Kullanıcı.xsd"/>
<xsd:include schemaLocation="Grup.xsd"/>
```

Böylece yukarıda belirlenen herbir elemana ilişkin özelliklerin gösterimi ile ilgili xsd. dökümanı oluşturulmaktadır. Ancak bu kısımda sadece "Haklar" elemanına ilişkin yapı aşağıda örneklendirme amacı ile gösterilmektedir. Burada, kullanıcıya verilen hakların "Okuma", "Yazma", "Değiştirme" olacağı gösterilmekte ve ilgili kısıtlamalar ve tanımlamalar yapılmaktadır.

```
<xsd:element name="Haklar">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element ref="Haklar "
maxOccurs="unbounded"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
<xsd:element name="Haklar">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="Yazma"/>
```

```
            <xsd:element ref="Okuma"/>
            <xsd:element ref="Degistirme"/>
          </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
<xsd:element name="Haklar">
 <xsd:sequence">
     <xsd:element name="Yazma" type="xsd:integer"/>
     <xsd:element name="Okuma" type="xsd:string"/>
     <xsd:element name="Degistirme" type="xsd:string"/>
  </xsd:sequence">
</xsd:element name="Haklar">
</xsd:schema>
```

Böylece veritabanlarına ilişkin güvenlik politikaları için ortaya konan yapı oluşturulan XML-Şema dökümanı ile ifade edilerek örneklendirilmiştir.

6. SONUÇ

Yapılan çalışmada veritabanı uygulamaları için zorunlu olan güvenliğin ancak organizasyonlar içinde tanımlanmış politikalar ile sağlanabileceği açıklanmaktadır. Bunun için iki örnek politika oluşturulmuş ilgili politikalar için belirlenen metaveri modelleri çizilmiştir. Anılan çizimlerin bir uygulama ile ifade edilebilmesi için XML- Şema kullanılmış ve her modele ait eleman yapıları ortaya konmuştur.

Sonuç olarak, her XML-Şema dökümanı aynı zamanda bir XML dökümanıdır. XML kullanımının sağladığı, genişletilebilirliği gösteren "modülerlik" ve diğer XML dökümanlar ile uyumu sağlayan "interoperability" kavramları yapılan uygulamada gösterilmektedir. Böylece hem metaveri tasarımı ile veri modellerinin taşınması, hem de bir organizasyon içinde belirlenen politikaların, değişen durumlar karşısında yeniden düzenlenmesi kolaylaşacak, metaveriler üzerinde yapılacak değişiklikler ile, ilgili politikaların süreç içinde devamlılığı sağlanacaktır.

7. KAYNAKLAR

Al-Qahtani, Abdullah, 2000, "Database Security", Computer Security Concepts Electronic Journal,Volume 1, Number 1.

Berners, T., Fielding Lee, R., Masinter, L., 1998. IETF (Internet Engineering Task Force) Uniform Resource Identifiers (URI): Generic Syntax, eds., RFC2396.

Doruk, Alpay, 2001. "BT Şirketlerinde Prensip Tabanlı Bilgi Güvenliği Çalışmaları", Bilişim2001.

Euardo B. Fernandez, Rouyi Pan, 2001. "A Pattern Language for Security Model", 8th Conference on Pattern Languages of Programs.

Henning, R. Ronda, Corporation Harris, 1996. "Use of the Zachman Architecture for Security Engineering", Information Systems Division.

Tannenbaum, Adrienne, 2001. "Metadata Solution", ISBN 0-201-71976-2, Sayfa 91, 160.

W3C Recommendation, 2000. "Extensible Markup Language (XML) 1.0 (Second Edition)", http://www.w3c.org.

W3C Recommendation, 2001,"XML Schema Part 1: Structures", http://www.w3.org/TR/xmlschema-1/

W3C Recommendation, 2001,"XML Schema Part 2: Datatypes" http://www.w3.org/TR/xmlschema-2/.