2004－CSEC－27　(16)
2004／12／20

セキュリティ運用管理のためのポリシー言語 SCCML

岡城純孝　松田勝志　小川隆一
ＮＥＣインターネットシステム研究所

要旨

インターネットに対する様々な脅威からネットワークを保護するために、ネットワーク全体のセキュリティの一貫性を保ちつつ統一的にネットワークセキュリティを実現する方法が求められている。現在、いろいろなセキュリティ機能を持った多種多様なセキュリティ機器が存在しているため、それらの設定･管理は非常に複雑であり、困難である。そこで我々は、管理者の負担を軽減し設定ミスを防止するため、ポリシーによるセキュリティ運用管理を行うシステムの研究を行っている。

本稿では、様々なセキュリティ機器の設定情報を統一的に表現可能なポリシー言語 SCCML を提案する。SCCML は、(1)アクセス制御と監視を行うセキュリティ機器のポリシーを表現できる、(2)複数の機能を持つセキュリティ機器のポリシーを表現できる、(3)実際のセキュリティ機器の設定情報を表現できる具体性を持つことを特徴とする。

A Policy Description Language for Policy-based Security Management

Sumitaka OKAJO, Katsushi MATSUDA and Ryuichi OGAWA

Abstract

In order to protect networks against network security threats, many security components with various security functions have been deployed, and the configuration and management of those components are highly complex. Therefore, we need a security policy management system to reduce system administrator's load.

This paper presents a common security policy language, SCCML(Security Configuration Coordinator Markup Language), which can express various configuration of security components. SCCML has three features; (1)to express both access control and monitoring policies, (2)to express multiple security functions in a policy and (3)to express detailed of a specific device.

１．はじめに

インターネットの普及に伴い、個人法人を問わずインターネットが重要な通信インフラとして認知されてきている。例えば企業では、社内情報のやり取りに電子メールや電子掲示板が必要不可欠となっているし、社内外の情報提供にはWWWが使われ、企業間の取引にもWebサービスが使われるようになってきている。

その一方で、インターネットに対する脅威も非常に高まっている。電子メールなどを介したコンピュータウイルスやワームによる被害、WWWサーバをはじめとするサーバへの不正侵入、コンテンツの改ざんなどである。これに対して、各企業ではファイアウォール、ウイルス対策ソフト、IDS（Intrusion Detection System, 侵入検知システム）などのセキュリティ関連のハードウェアやソフトウェアを積極的に導入することによりセキュリティを確保しようとしている。

しかしながら、複数ベンダが提供する様々な種類のハードウェア／ソフトウェアによって構成されるネットワークにおいて、すべてのセキュリティ設定を個別に正しく行うことは優秀な管理者であっても非常に難しく、負担も大きい。

そこで現在、管理者の負担を軽減するセキュリティ運用管理方式が必要とされており、我々はこれを実現するセキュリティポリシー管理基盤の研究を行っている。本稿では、様々なセキュリティ機器の設定情報を統一的に管理できるセキュリティポリシー記述言語を提案する。

２．セキュリティポリシー

一般にセキュリティポリシーは図１のような階層で表される[1]。

NEC インターネットシステム研究所
Internet Systems Research Laboratories. NEC Corp.

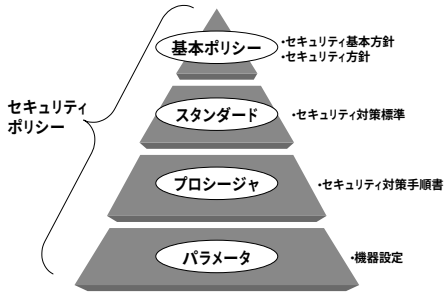

図 1　セキュリティポリシーの基本的な階層

- 基本ポリシー:情報セキュリティの方針を宣言したもの
- スタンダード:基本ポリシーに従い、必要な対策を分野別に記述もの
- プロシージャ:スタンダードを実施するための詳細な手順を記述したもの
- パラメータ:プロシージャを機器設定にしたもの

通常、セキュリティポリシーの適用は、基本ポリシーからスタンダードへ、スタンダードからプロシージャへとブレークダウンしていき、最後に管理者がプロシージャの内容を満足するように各セキュリティ機器にパラメータを設定することにより実現される。

本稿で扱うセキュリティポリシーは、プロシージャおよびパラメータの範囲を対象とする。以降、単にポリシーとした場合はプロシージャを、ポリシー記述とした場合はセキュリティ機器のパラメータを表す。

３．問題とアプローチ

本節では、まず、セキュリティ運用管理の現状と問題について述べ、次にその問題を解決するためのアプローチについて述べる。

３．１．セキュリティ運用管理の現状と問題

各組織では、情報システムを攻撃や不正侵入から守る目的で様々なカテゴリのセキュリティ機器が導入されている。また、同一のカテゴリに属するセキュリティ機器であっても、異なるベンダのセキュリティ機器が１つの情報システム内で導入されていることも多い。さらに、情報システムの日々の運用において各セキュリティ機器の設定は変更されていく。

このようなセキュリティ運用管理の現状では以下のような問題がある。

- **ポリシー表現の不統一**

同一のカテゴリに属するセキュリティ機器であってもベンダが異なればポリシーの記述方法も異なる。また、同じベンダのセキュリティ機器であっても機種やバージョンが異なればポリシーの記述方法が異なることもある。

- **管理者の負担の増大**

前記の問題に付随して、セキュリティ機器の設定を行う際に管理者は、ベンダごと、および機種ごとのポリシー記述方法を理解した上で設定を行わなければならない。例えば、ベンダＡのファイアウォール FW(A)とベンダＢのファイアウォール FW(B)に同じポリシーを設定しようとする場合に、管理者はそのポリシーの内容を解釈しFW(A)と FW(B)のポリシー記述に変換したうえで、それぞれにポリシーを設定する必要がある。このため管理者の負担が増大するだけでなく、設定ミスが生じる可能性も高くなる。さらに、セキュリティ機器の故障やバージョンアップに伴う機器のリプレースを行わなければならない場合にも同様の問題が生じる。

３．２．アプローチ

前述した問題を解決するために、本研究では、セキュリティ機器に依存しないセキュリティポリシーのモデルおよび記述言語を構築する方針を採り、以下のようなステップでこれを実現した。

（１）　セキュリティ機器の動作のモデル化

セキュリティ機器に依存しないセキュリティポリシーのモデルおよび記述言語を構築するために、まずはセキュリティ機器が持つセキュリティ機能の動作と、それによって制御されるオブジェクトからなる動作モデルの検討を行った。ここでセキュリティ機能とは、各セキュリティ機器が行う動作におけるセキュリティに関する最小単位の処理を指す。例えばファイアウォールでは、パケットフィルタリングやアドレス変換がセキュリティ機能に相当する。セキュリティ機器の動作を抽象化して表現することで、機器固有のポリシーの意味と目的の明確化、および類似セキュリティ機器との関連性の明確化を図った。

この動作モデルの検討によって、まず、セキュリティ機器をアクセス制御系機器と監視系機器の２つに大別した。

（２）　セキュリティポリシーのモデル化

前述のアクセス制御系機器と監視系機器では、動作の振る舞いや入出力が異なる。よって次に、それぞれの動作モデルについて、機能の種類、入出力オブジェクトの種類、動作環境の種類などを列挙・整理することでセキュリティポリシーのモデル化を行った。

（３）　セキュリティポリシー記述言語の構築

最後に、前記セキュリティポリシーのモデルに基づくポリシー言語を構築した。

４．セキュリティ機器動作モデル

本節では、アクセス制御系機器および監視系機器の動作モデルについて述べる。

４．１．アクセス制御系動作モデル

アクセス制御系機器には、パケットフィルタリングを行うファイアウォールやルータ、ネットワークサービスへのアクセス制御を行うスーパーサーバなどが含まれる。これらアクセス制御系機器の動作モデルは図２のように表すことができる。

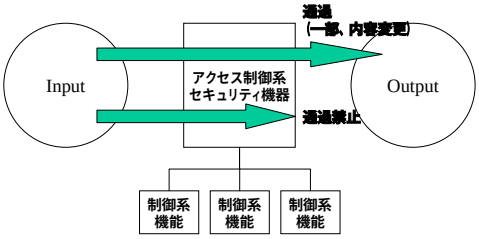

図 2　アクセス制御系機器の動作モデル

アクセス制御系機器の動作は、

① 入力オブジェクト Input の通過を許可あるいは拒否する

② 入力オブジェクト Input を出力オブジェクト Output に変換する

の２通りに集約される。前者の代表例はパケットフィルタリングによるアクセス制御であり、後者の代表例はアドレス変換である。アドレス変換では入力パケットの IP アドレスやポート番号を書き換えたパケットが出力される。

また、セキュリティ機器によっては複数の機能を組み合わせることにより動作するものが存在する。例えば、パケットフィルタリングと URL フィルタリングを組み合わせることにより Web サーバへのアクセス制御を行うようなファイアウォールなどがある。

４．２．監視系動作モデル

監視系機器には、侵入検知を行う IDS や不正なファイル書き換えを監視するファイル改ざん監視ソフトなどが含まれる。これら監視系機器の動作モデルは図３のように表すことができる。監視系機器は、監視対象オブジェクト Object を監視し、該当する属性を持つオブジェクトが検知されたときに反応Responseを出力するように動作する。

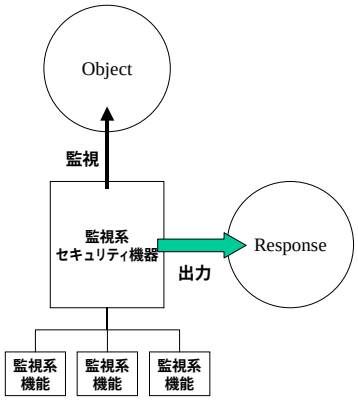

図 3　監視系機器の動作モデル

５．セキュリティポリシーモデル

本節では、前述の動作モデルに基づいたセキュリティポリシーモデルについて述べる。

５．１．アクセス制御系ポリシーモデル

アクセス制御系機器のポリシーには、個々のポリシーに順序関係が存在しない場合と存在する場合がある。本稿ではこれらを２つに分け、前者を集合型ポリシー、後者をリスト型ポリシーと呼ぶ。例えば、スーパーサーバでのアクセス制御はサービスごとに規定され、ポリシーに順序関係は存在しないので集合型ポリシーとなる。一方、パケットフィルタリングは個々のポリシーの順序関係によってポリシー全体の意味が変化するのでリスト型ポリシーとなる。

アクセス制御系ポリシーモデルは図４のように表すことができる。

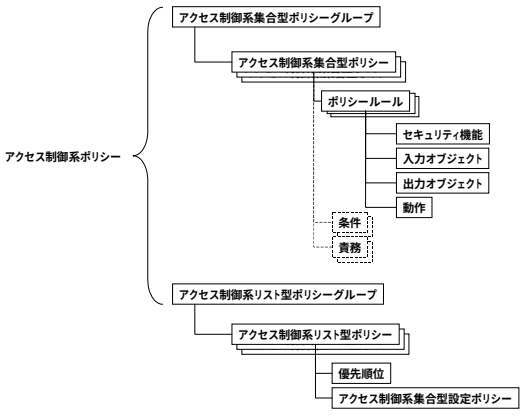

図 4　アクセス制御系ポリシーモデル

ある１つのアクセス制御系機器のポリシー全体はアクセス制御系集合型ポリシーグループあるいはアクセス制御系リスト型ポリシーグループのいずれかになる。ポリシーグループはそれぞれのタイプのポリシーを１つ以上含む。

アクセス制御系集合型ポリシーは１つ以上のポリシールールを含む。ポリシールールはセキュリティ機能単位で規定する。ポリシールールは、セキュリティ機能、セキュリティ機能への入力オブジェクト、セキュリティ機能の動作、セキュリティ機能からの出力オブジェクトからなる。例えば、ファイアウォールの持つセキュリティ機能であるパケットフィルタリングに関する個々のポリシールールは、セキュリティ機能が「パケットフィルタリング」、セキュリティ機能への入力オブジェクトが「パケット型オブジェクト」、セキュリティ機能の動作が「通過許可」あるいは「通過禁止」、出力オブジェクトが「(入力オブジェクトと同一の）パケット型オブジェクト」（通過許可の場合）となる。また、前述したように複数のセキュリティ機能を組み合わせて動作する機器の場合には、１つのアクセス制御系集合型ポリシーの中にそれぞれのセキュリティ機能に対応する複数のポリシールールが含まれることになる。

さらに、アクセス制御系集合型ポリシーはオプションとして条件と責務を含むことができる。ポリシーが条件を含む場合には、この条件が満足される場合にのみポリシーが評価される。例えば、「イントラネットからインターネットへの Web アクセスを許可する」という意味を持つパケットフィルタリングポリシーに、「午前９時から午後５時まで」という条件を与えることができる。また、ポリシーが責務を含む場合には、ポリシー適用時に責務で指定した処理が実行される。例えば、「インターネットから DMZ 内への Web アクセスを許可する」という意味を持つパケットフィルタリングポリシーに、「ログを残す」という責務を与えることができる。

一方、アクセス制御系リスト型ポリシーは、前記アクセス制御系集合型ポリシーに各々のポリシーの優先順位を付加したものとなる。アクセス制御系リスト型ポリシーはこの優先順位に従って評価される。

５．２．監視系ポリシーモデル

一般に、監視系機器の設定においては個々のポリシーに順序関係は存在しないため、監視系ポリシーは集合型ポリシーとなる。

監視系ポリシーモデルは図５のように表すことができる。

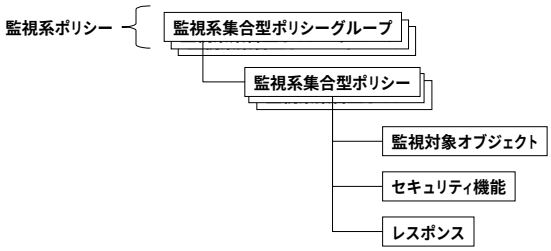

図 5　監視系ポリシーモデル

アクセス制御系ポリシーグループはアクセス制御系セキュリティ機器と１対１に対応していたが、監視系ポリシーグループはセキュリティ機能と１対１に対応する。複数の機能を組み合わせて動作する監視系機器の場合には、それぞれの機能ごとにポリシーグループを規定する。

監視系集合型ポリシーグループは１つ以上の監視系集合型ポリシーを含む。監視系集合型ポリシーは、セキュリティ機能、監視対象となるオブジェクト、監視によって該当オブジェクトが検知されたときのレスポンスからなる。例えば、監視系セキュリティ機器の１つであるシグネチャ型 NIDS のポリシーは、セキュリティ機能が「パケット監視」、監視対象オブジェクトが「パケット型オブジェクト」、レスポンスが「アラートオブジェクト」（Email による通知など）となる。

６．SCCML

本節では、前節で述べたモデルに基づき構築したセキュリティポリシー記述言語 SCCML(Security Configuration Coordinator Markup Language)について述べる。SCCML は標準化団体 OASIS(Organization for the Advancement of Structured Information Standards)が制定したインターネット経由の情報アクセスに関する制御ポリシーを表現する XACML(eXtensible Access Control Markup Language)[2]をベースに拡張した XML 形式の言語である。

６．１．アクセス制御系ポリシー記述

アクセス制御系ポリシーモデルに基づいて構築した SCCML の主な要素を図６に示す。

Policy 要素が個々のアクセス制御ポリシーと１対１に対応する。Policy 要素は、ポリシールールを表す１つ以上の PolicyRule 要素と、条件を表す０以上の Condition 要素と、責務を表す Obligation 要素を含む。ポリシーグループを表す PolicyGroup 要素はセキュリティ機器と１対１に対応し、複数の Policy 要素を含む。さらに、複数のセキュリティ機器のポリシーをグループ

化できるように PolicySet 要素を用意した。PolicyRule 要素は、セキュリティ機能を表す Function 要素、入力オブジェクトを表す InputObject 要素、出力オブジェクトを表す OutputObject 要素、セキュリティ機能の動作を表す Action 要素からなり、この４つの要素を Target 要素でまとめている。

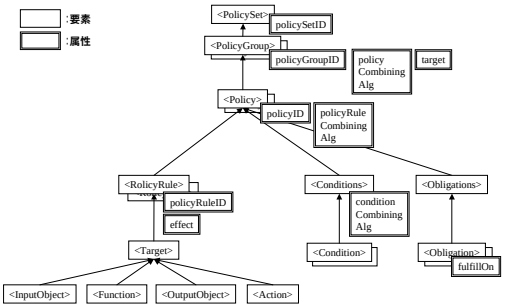

図 6　SCCML の構造(アクセス制御系ポリシー)

PolicyGroup 要素の policyCombiningAlg 属性は、ポリシーグループが集合型かリスト型かを表す。この属性でリスト型が指定されている場合には、この PolicyGroup 要素が含む Polciy 要素は、先頭から高い優先順位を持つものとして扱われる。一方、集合型が指定されている場合には、Policy 要素の順序は意味を持たない。また、target 属性は”firewall”などセキュリティ機器の種類を表す。

PolicyRule 要素の effect 属性は、この PolicyRule 要素内の Action 要素で表される動作を許可する(permit)か禁止する(deny)かを表す。

Policy 要素の policyRuleCombiningAlg 属性は、この Polciy 要素に含まれる複数の PolicyRule 要素の評価の結合アルゴリズムを表す。属性値”ordered-deny-overrides”が指定されている場合には、少なくとも１つの PolicyRule 要素の effect 属性が”deny”と判断されると親の Policy 要素の評価を”禁止(deny)”とする。また、属性値”ordered-permit-overrides”が指定されている場合には、少なくとも１つの PolicyRule 要素の effect 属性が”permit”と判断されると親の Policy 要素の評価を”許可(permit)”とする。

Conditions 要素の conditionCombiningAlg 属性に”and”が指定されている場合には、この Conditions 要素が含む Condition 要素をすべて満足する場合にのみ、親の Policy 要素を適用することを表す。また、”or”が指定されている場合には、この Conditions 要素が含む Condition 要素のうち少なくとも１つを満足する場合に、親の Policy 要素の内容を適用することを表す。

Obligations 要素の fulfillOn 属性は Obligation 要素の実行条件を表す。属性値として”permit”あるいは”deny”が指定可能で、この値と、親の Policy 要素の評価結果とが一致した場合にすべての Obligation 要素の内容が実行される。

図７に SCCML によるアクセス制御系ポリシーの記述例の一部を示す。これは、責務としてログを残す(Track)ことを指定したパケットフィルタリングポリシーの一例である。

```
<Policy policyID="contentsSecurity001" policyRuleCombiningAlg="ordered-deny-overrides">
  <PolicyDescription>LANから外部のWebサーバへの接続を許可する</PolicyDescription>
  <PolicyRule policyRuleID="packetFiltering001" effect="permit">
    <PolicyRuleDescription>LANから外部のWebサーバへのアクセス</PolicyRuleDescription>
    <Target>
      <Function>packet_filtering</Function>
      <InputObject>
        <Packet>
          <SrcIP>192.168.1.0/32</SrcIP>
          <SrcPort>any</SrcPort>
          <Protocol>tcp</Protocol>
          <DestIP>0.0.0.0/0</DestIP>
          <DestPort>80</DestPort>
        </Packet>
      </InputObject>
      <Action>accept</Action>
      <OutputObject/>
    </Target>
  </PolicyRule>
  <Obligations fulfillOn="permit">
    <Obligation>
      <Track>long</Track>
    </Obligaion>
  </Obligations>
</Policy>
```

図 7　SCCML 記述例(アクセス制御系ポリシー)

６．２．監視系ポリシー記述

監視系ポリシーモデルに基づいて構築した SCCML の主な要素を図８に示す。

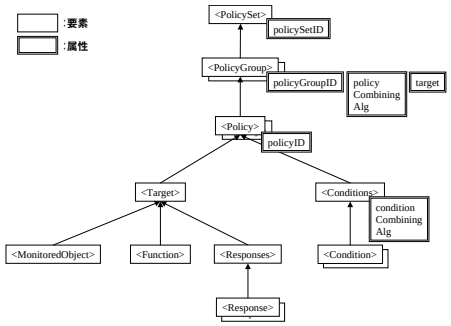

図 8　SCCML の構造(監視系ポリシー)

Policy 要素が個々の監視系ポリシーと１対１に対応する。Policy 要素は監視対象オブジェクトを表す MonitoredObject 要素、セキュリティ機能を表す Function 要素、レスポンスを表す Response 要素からなり、この３つの要素を Target 要素でまとめている。その他の要素および属性についてはアクセス制御系ポリシーの場合と同様である。

図９に SCCML による監視系ポリシーの記述例の一部を示す。これは、レスポンスとして Email による通知を指定したシグネチャ型 NIDS ポリシーの一例である。

```
<PolicyGroup policyGroupID="nidsPolicy001"
        policyCombiningType="independent">
        target="nids">
   <PolicyGroupDescription>NIDSに関するポリシー</PolicyGroupDescription>
   <Policy policyID="packetMonitoring001">
      <PolilyDescription>subsevenバックドアを監視する</PolicyDescription>
      <Target>
         <MonitoredObject>
            <SecurityEvent>
               <EventName>backdoor, subseven</EventName>
            </SecurityEvent>
         </MonitoredObject>
         <Function>
            <PacketMonitoring>
               <Enabled>true</Enabled>
               <Priority>high</Priority>
            </PacketMonitoring>
         </Function>
         <Responses>
            <Response>
               <Email>
                  <MailServer>192.168.3.10</MailServer>
                  <MailAddress>abc@xxx.com</MailAddress>
               </Email>
            </Response>
         </Responses>
      </Target>
   </Policy>
</policygroup>
```

図 9 SCCML 記述例（監視系ポリシー）

６．３．記述力の検証

これまでに我々は SCCML を用いて、セキュリティ製品の中で最も記述力が高いと考えられる CheckPoint 社の FireWall-1 のアクセス制御系ポリシーや、ISS 社の RealSecure NetworkSensor の監視系ポリシーを記述できることを確認した。SCCML は、少なくともファイアウォール製品と NIDS 製品のポリシーについては充分な記述力を持つと考える。

７．関連研究との比較

本節では、代表的なポリシー記述言語を幾つか挙げて SCCML との比較を行う。

- **PCIM(Policy Core Information Model)**[3]

PCIM は、一般的なポリシー情報を表現するためのオブジェクト指向モデルであり、特定コンディション(PolicyCondition クラス)下でのアクション(PolicyAction クラス)を規定するルール(PolicyRule クラス)またはその集合(PolicyGroup クラス)をポリシーとしている。

PCIM はセキュリティに限らず QoS や暗号化などネットワークにおける様々な制御／管理で共通に使用されるポリシーのフレームワークのみを規定しているに過ぎず、実運用で用いるためには新たな概念や具体的なクラスを追加しなければならない。これに対して、SCCML は実際のセキュリティ機器の動作やそれによって制御されるオブジェクトに基づいてモデル化を行っているため、実運用に充分な具体性を持つ。

- **XACML**

XACML の目的は、インターネット経由の情報アクセス許可ポリシー用の統一言語を提供し、多種多様なアクセス管理／許可機器の相互接続を実現することである。つまり、“特定の人物によるファイルアクセスの許可／禁止を決めるルール”を表現するための統一言語と言える。

しかしながら、アクセス制御系セキュリティ機器の動作には、アドレス変換に代表されるような XACML では表現できないポリシーが存在する。SCCML の記述力は XACML を内包しているので、このようなポリシーも表現することができる。

- **Ponder**[4]

Ponder は分散システム管理のためのポリシー記述言語であり、柔軟で拡張可能なオブジェクト指向言語である。Ponder では、アクセス制御のための Authorization Policy と、イベントトリガーの条件とアクションの if-then 形式で表現される Obligation Policy などが規定されている。

Ponder はアクセス制御系ポリシーを記述する一般的な方法を提供する非常に柔軟な言語であり、大規模な企業情報システムにも対応できるように設計されている。しかし、SCCML で可能な、複数の機能を組み合わせたポリシーの記述は Ponder では困難である。また、監視系ポリシーについては想定されていない。

８．おわりに

本稿では、様々なセキュリティ機器の設定情報を統一的に管理できるセキュリティポリシー記述言語 SCCML について述べた。SCCML の構築にあたり、まずセキュリティ機器の動作のモデル化を行い、次にその動作を規定するポリシーをモデル化し、最後にポリシーモデルに基づいたポリシー記述言語を構築したことについて述べた。また、SCCML と他のポリシー記述言語との比較を行った。

現在、SCCML を用いることにより、セキュリティ機器間のポリシーを比較したり、整合性の確認を可能にするセキュリティポリシー管理ミドルウェアを開発中であり、SCCML およびミドルウェアの有効性を検証していく予定である。

【参考文献】

[1] 情報セキュリティポリシー・サンプル 0.92a 版, http://www.jnsa.org/policy/guidance/.

[2]OASIS , XACML Version1.1 , http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf.

[3] IETF Policy Framework WG , PCIM – Version1 Specification(RFC3060), http://www.ietf.org/rfc/rfc3060.txt.

[4]Nicodemos Damianou, Naranker Dulay, Emil Lupu, Morris Sloman , ”The Ponder Policy Specification Language”, Proc. Policy2001.