Mr Khiêm Support

Em chào anh, em xin tranh thủ thời gian anh rảnh rỗi nhờ anh xem qua giúp em luôn ạ, em xin nói tóm tắt như sau cho dễ hiểu ạ.

Hiện tại em cần một chương trình đo đạc tính định tính và tính định lượng của em ạ.

Em sẽ phân ra làm 3 phần nói cho đơn giản ạ

**Phần 1:** Phân tích từ cấu trúc XML ban đầu thành danh sách liên kết các attribute types (at) và applicable constraint (ac)

(at) là nơi lưu các thông tin về loại thuộc tính và các thông tin liên quan đến ngữ cảnh (thời gian, địa điểm, vai trò, …) lưu ý là không phải giá trị nha anh

(ac) là nơi lưu các thông tin về điều kiện ràng buộc hay giá trị ràng buộc (lớn hơn, nhỏ hơn, khác, bằng) …

Em xin lấy ví dụ trong thằng Continue-a cho dễ hiểu luôn ạ

Chính sách này sẽ được khai báo dưới định dạng XML cho XACML v3.0 như sau:

**<PolicySet PolicySetId="RPSlist.0" PolicyCombiningAlgId = "urn:oasis:names:tc:xacml: 1.0:policy-combining-algorithm:first-applicable">**

**<Target>**

**<Resources>**

**<Resource>**

**<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">**

**<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">confere nce\_rc</AttributeValue>**

**<ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0: resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>**

**</ResourceMatch>**

**</Resource>**

**</Resources>**

**</Target>**

**<PolicySet PolicySetId="RPSlist.0.0"PolicyCombiningAlgId="urn:oasis:names:tc:xacml :1.0:policy-combining-algorithm:first-applicable">**

**<Target/>**

**<Policy PolicyId="RPSlist.0.0.0" RuleCombiningAlgId="urn:oasis:names:tc:xacml: 1.0:rule-combining-algorithm:first-applicable">**

**<Target>**

**<Subjects>**

**<Subject>**

**<SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">**

**<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"> admin</AttributeValue>**

**<SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc: xacml:1.0:subject-category:access-subject" AttributeId="role" DataType="http://www.w3.org/ 2001/XMLSchema#string"/>**

**</SubjectMatch>**

**</Subject>**

**</Subjects>**

**<Actions>**

**<Action>**

**<ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">**

**<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"> read</AttributeValue>**

**<ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0 :action:action-id" DataType="http://www.w3.org/2001/XMLSchema #string"/>**

**</ActionMatch>**

**</Action>**

**<Action>**

**<ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">**

**<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"> write</AttributeValue>**

**<ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0: action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>**

**</ActionMatch>**

**</Action>**

**</Actions>**

**</Target>**

**<Rule RuleId="RPSlist.0.0.0.r.1" Effect="Permit">**

**<Target/>**

**</Rule>**

**</Policy>**

**<Policy PolicyId="RPSlist.0.0.1" RuleCombiningAlgId="urn:oasis:names:tc:xacml: 1.0:rule-combining-algorithm:first-applicable">**

**<Target>**

**<Subjects>**

**<Subject>**

**<SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">**

**<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">pc-chair</AttributeValue>**

**<SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml :1.0:subject-category:access-subject" AttributeId="role" DataType="ht tp://www.w3.org/2001/XMLSchema#string"/>**

**</SubjectMatch>**

**</Subject>**

**</Subjects>**

**<Actions>**

**<Action>**

**<ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">**

**<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"> read</AttributeValue>**

**<ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0: action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>**

**</ActionMatch>**

**</Action>**

**</Actions>**

**</Target>**

**<Rule RuleId="RPSlist.0.0.1.r.1" Effect="Permit">**

**<Target/>**

**</Rule>**

**</Policy>**

**<Policy PolicyId="RPSlist.0.0.2" RuleCombiningAlgId="urn:oasis:names:tc:xacml: 1.0:rule-combining-algorithm:first-applicable">**

**<Target>**

**<Subjects>**

**<Subject>**

**<SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">**

**<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">pc-member</AttributeValue>**

**<SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml: 1.0:subject-category:access-subject"AttributeId="role"DataType="http: //www.w3.org/2001/XMLSchema#string"/>**

**</SubjectMatch>**

**<SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">**

**<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">true </AttributeValue>**

**<SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml: 1.0:subject-category:access-subject" AttributeId="isMeeting"DataType= "http://www.w3.org/2001/XMLSchema#string"/>**

**</SubjectMatch>**

**</Subject>**

**</Subjects>**

**<Actions>**

**<Action>**

**<ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">**

**<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"> read</AttributeValue>**

**<ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0: action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>**

**</ActionMatch>**

**</Action>**

**</Actions>**

**</Target>**

**<Rule RuleId="RPSlist.0.0.2.r.1" Effect="Permit">**

**<Target/>**

**</Rule>**

**</Policy>**

**<Policy PolicyId="RPSlist.0.0.3" RuleCombiningAlgId="urn:oasis:names:tc:xacml: 1.0:rule-combining-algorithm:first-applicable">**

**<Target/>**

**<Rule RuleId="RPSlist.0.0.3.r.1" Effect="Deny">**

**<Target/>**

**</Rule>**

**</Policy>**

**</PolicySet>**

**</PolicySet>**

Trong Continue-a thì nó chỉ có một policyset tổng thôi ạ. Em nhờ anh cắt thành từng cặp policyset ở mức thứ hai ạ. Đây là policyset đầu tiên của continue-a ạ. Em xin tóm tắt nội dung của Policy Set ở trên như sau ạ:

Nội dung chính của policy này là cung cấp quyền truy cập cho tập tin “**conference\_rc**” cho các nhóm đối tượng người dùng bao gồm ba vai trò {**admin, pc-chair, pc-member**}. Các quyền cho phép trên tập tin này bao gồm {**read, write**}. Với các yêu cầu ràng buộc như sau:

* Nhóm người dùng có vai trò là **admin** sẽ có toàn quyền {**read, write**} trên tập tin **conference\_rc.**
* Nhóm người dùng có vai trò là **pc-chair** sẽ có quyền đọc {**read**} trên tập tin **conference\_rc.**
* Nhóm người dùng có vai trò là **pc-member** chỉ có quyền đọc {**read**} trên tập tin **conference\_rc** khi và chỉ khi đang trong cuộc họp (**isMeeting = true**).

Các thuộc tính của chính sách mẫu:

* Effect: “Permit”
* Combining algorithm: “first-applicable”
* Thể hiện chính sách: “Permit”

Phân tích thành danh sách các at và ac bằng các thuật toán liên kết. Dạ tất cả đều theo một giải thuật kết hợp cho cả ba dữ liệu mẫu (cả GEYSERS và KMarket cũng vậy ạ)

**PolicySet** (**Effect** là Permit – **Combining Algorithm** là First Applicable)

{

Xác định vùng dữ liệu tương tác (**resource**) và gán giá trị cho at0.

at0: **conference\_rc** resource-type

Nhóm người dùng có vai trò là **Admin** sẽ có toàn quyền trên tập tin **conference\_rc**

PolicySet (**Effect** là Permit – **Combining Algorithm** là First Applicable) (1)

{

at1: Role

at2: action-id

}

{

ac0: **admin** Role

ac1: **read** action-id

ac2: **write** action-id

}

Nhóm người dùng có vai trò là **pc-chair** sẽ có toàn quyền **read** trên tập tin **conference\_rc**

Policy (**Effect** là Permit – **Combining Algorithm** là First Applicable) (2)

{

at1: Role

at2: action-id

}

{

ac3: **pc-chair** Role

ac4: **read** action-id

}

Nhóm người dùng có vai trò là **pc-member** sẽ có toàn quyền **read** trên tập tin **conference\_rc** khi và chỉ khi **isMeeting** = "true"

Policy (**Effect** là Permit – **Combining Algorithm** là First Applicable) (3)

{

at1: Role

at2: action-id

}

{

ac5: **pc-member** Role

ac6: **isMeeting** = "true"

ac7: **read** action-id

}

Ngoài ba điều kiện trên tất cả yêu cầu của người dùng đều bị “Deny”

Policy (**Effect** là Deny – **Combining Algorithm** là First Applicable) (4)

{

ac8: **Deny**

}

Xây dựng nhóm ac xác định vi phạm Indeterminate. Trong trường hợp bất kỳ câu request nào từ phía người dùng không chứa các điều kiện ràng buộc này sẽ trả về kết quả “Indeterminate”. Việc quản lý một ac được tạo mới trong quá trình cập nhật sẽ được quản lý thông qua chính sách tổng (policy set).

{

at0: **Resource-type**

at1: **Role**

at2: **action-id**

at3: **isMeeting**

}

}

Lưu ý: trường hợp at0 mà khác thì sẽ trả về Not Applicable ạ. Nhận biết đơn giản lắm ạ. Nó sẽ nằm ở policy định nghĩa đầu tiên luôn cho toàn bộ Policy Set của nó ạ

Chính sách tổng ban đầu sẽ bao gồm 4 tập policy con kết hợp thông qua giải thuật kết nối **First Applicable** như sau First Applicable{(1), (2), (3), (4)}. Với:

(1) : <D, P, IN, NA>

D : Empty

P : or (and (ac0; ac2); and (ac0; ac3))

IN: or (at1; at2)

NA: not (at0)

(2) : <D, P, IN, NA>

D : Empty

P : and (ac3; ac4)

IN: or (at1; at2)

NA: not (at0)

(3) : <D, P, IN, NA>

D : Empty

P : and (ac5; ac6; ac7)

IN: or (at1; at2; at3)

NA: not (at0)

(4) : <D, P, IN, NA>

D : true (ac8)

P : Empty

IN: Empty

NA: Empty

**Lưu ý**: ở luật bảo mật số (4) giá trị Deny luôn trả về là ***true*** nhằm thể hiện trường hợp yêu cầu truy cập không thỏa mãn bất kỳ ràng buộc nào có trong luật bảo mật (1), (2) và (3). Điều này tương đương với Deny(4)***true*** = Deny(1) Deny(2) Deny(3)

Policy tổng ban đầu sẽ có giá trị như sau:

|  |  |
| --- | --- |
| Permit: | Deny: |
| Not Applicable: | Indeterminate: |

Với:

Giá trị at0

Giá trị (ac0 ac1 ac2) (ac3 ac4) (ac5 ac6 ac7)

Giá trị at1 at2at3

**Cách biểu diễn đầu ra**

Ký hiệu và/giao (): and(,)

Ký hiệu hoặc/hợp (): or(,)

Ký hiệu not (): not()

Dạ như vậy là xong phần thứ nhất rồi ạ. Nghĩa là mình sẽ cho vào các tập tin định dạng file xml ạ. Đầu ra sẽ là chuỗi các at và ac ạ. Phần này đợt trước Tuấn Anh có hỗ trợ em rồi ạ. Nhưng do em muốn tách đôi thành các ac và at cho dễ dàng quản lý ạ. Ví dụ: đối với cập nhật dạng at thì chỉ có xoá và thêm (không có chỉnh sửa); còn với at thì có cả 3 bao gồm thêm, sửa và xoá ạ. Khi xoá at thì không làm gì cả còn thêm các at thì sẽ xét xem có cần viết lại câu truy vấn hay không ạ. At thì tương tự như đợt trước em nói với anh thôi ạ.

Code của Tuấn Anh em sẽ gửi cho anh để anh chỉ customize thôi ạ. (Phần này em nghĩ là làm sẽ nhanh ạ). Còn phần chuyển đổi câu truy cấn thì Tuấn Anh làm luôn rồi ạ. Trong tập tin e chuyển anh thì có phần chuyển đổi câu truy cấn luôn ạ. Nó rất đơn giản chỉ là một tập các cặp thuộc tính và gí trị đi cùng thôi ạ. Cái này a không phải làm ạ, anh chỉ cần xem qua code cho hiểu giúp em thôi ạ.

Phần tìm chính sách (policy evaluation): Phần này rất quan trọng ạ vì bước này được xem như là đầu tiên luôn ạ, em chỉ quan tâm đến việc cập nhật chính sách trên cơ sở là chính sách đó đã cấp quyền cho yêu cầu nào hay chưa thôi ạ. Cách tìm cũng khá đơn giản á anh chủ yêu là so khớp và so sánh giá trị lớn hơn hay nhỏ hơn thôi ạ.

Cách duyệt sẽ theo kiểu thác nước á anh. Đầu tiên một yêu cầu sẽ được tách ra làm hai phần gồm các at và các ac (tương tự như chính sách ạ).

***So sánh các at của yêu cầu với at của chính sách***: Nếu sai các at đầu tiên (trong danh sách Not Applicable) thì trả về Not Applicable ạ. Nếu có các at trong danh sách Not Applicable mà thiếu các at khác thì sẽ trả về Indeterminate. Nếu trả về hai cái này thì sẽ đi đến danh sách kế tiếp ạ.

***So sánh với các ac của yêu cầu với ac của chính sách***: cái này phân làm hai trường hợp ạ. Trường hợp đầu tiên là sai một trong các ac của danh sách thì sẽ trả về giá trị ngược lại lưu ở Effect (là ac8 như ví dụ trên). Nếu đúng thì trả về giá trị lưu trữ ở Effect ạ (trả về giá trị ngược lại ac8).

Not(True) = False ⬄ Not(Permit) = Deny

Not(False) = True ⬄ Not(Deny) = Permit

**Phần 2:** thiết kế chiến lược đo đạc

Dạ trong phần này chú trọng đến việc thiết kế các chiến lược ạ. Hiện tại sau khi hoàn tất phần một mình chỉ cần quản lý các danh sách policy, at và ac thôi ạ. Mỗi giá trị sẽ lưu các kểu khác nhau anh có thể hỗ trợ em ngoài dạng:

policy(at0,…, atm, ac0,…, acn) Quản lý các at và ac có bên trong của nó (ở đây có thể ngầm hiểu là từng policy set ạ, như em ví dụ ở trên ạ mỗi policyset mức 2 trong continue-a sẽ được lưu trong một policy như vậy ạ).

at(loại thuộc tính): ví dụ role, current-time, current-place, age, name, address, … Nghĩa là ký tự đại diện cho một thuộc tính bất kỳ ạ. Lưu gồm danh sách các ac.

ac(ràng buộc điều kiện): các phép toán so sánh với một giá trị thuộc tính cụ thể ạ. Có thể là các phép toán equal, higher, less than, …. Lưu gồm một phép toán so sánh hay một biểu thức điều kiện ạ.

Ví dụ age > 18

Thì at là age và ac là > 18 ạ.

Đối với chiến lược thay đổi thì at chỉ có xoá hoặc là thêm thôi ạ. Tuy nhiên nếu như at mà xoá thì kéo theo nó là ac bị xoá theo ạ. Ví dụ ở trên at là age mà mất thì > 18 cũng mất luôn ạ. Mô phỏng giống như xoá policy hoặc cả một thuộc tính có trong policy vậy á anh.

Nhưng xoá điều kiện thì không bị ạ. Vì xoá ac thì sẽ ko ảnh hưởng đến at ạ. Ví dụ trến policy mẫu có nhiều giá trị action, em xoá ac1 thì nó vẫn không ảnh hưởng gì đến at2 hết ạ. Tương tự như xoá rule vậy á anh.

Anh có thể giúp em là một at sẽ có liên kết với các ac nào để khi xoá at sẽ xoá toàn bộ ac ạ. Giống kiểu danh sách vậy á anh.

at = {ac0, …, acn}

Cấu trúc đo đạc sẽ chia làm 3 phần ạ:

**Phần thứ nhất**: cập nhật bên trong chính sách cấp quyền truy cập

* **Xoá:**

Nếu xoá ac thì đi tiếp ạ.

Nếu xoá at thì xoá hết các ac đi kèm rồi cũng đi tiếp ạ

* **Cập nhật:**

Chia làm hai loại chính ạ.

Cập nhật không làm ảnh hưởng đến quá trình sinh lại chuỗi at và ac là cập nhât các giá trị thuộc tính, điều kiện ràng buộc => chỉ cần re-evaluate lại là đủ ạ

Cập nhật làm ảnh hưởng đến chuỗi các at và ac hiện tại nên phải tiến hành sinh lại chuỗi at và ac. Các cập nhật xảy ra ở thuật toán kết hợp (combining algorithm) và giá trị của effect.

* **Thêm:**

Đối với trường hợp thêm là at thì sẽ xem xét trong thông tin người dùng cung cấp đã có sẵn at mong muốn hay chưa nếu như chưa sẽ gửi một yêu cầu nhập giá trị của at tương ứng cho người dùng. Cái này a thiết kế giúp em dữ liệu trả về từ phía người dùng là dummy nha anh. Đối với việc viết lại câu truy vấn anh chỉ cần add thêm cho em phần giá trị thuộc tính và thuộc tính tương ứng vào câu truy vấn ban đầu thôi ạ. Nếu không có thông tin phản hồi (trả lời từ phía người dùng) trả về **Indeterminate**

Đối với trường hợp thêm là ac thì sẽ xem xét điều kiện thêm ạ. Có thể cập nhật lại điều kiện ràng buộc (Ví dụ age > 8 sau khi cập nhật là age > 20). Hoặc thêm một ac tương ứng (Ví dụ age > 8 sau khi cập nhật 8 < age < 20; hoặc khoa = “Công nghệ thông tin” sau khi cập nhật khoa = “Công nghệ thông tin” và khoa = “công nghệ thực phẩm”)

**Phần thứ hai:** cập nhật là xoá chính sách cấp quyền truy cập.

Ở đây anh giúp em một ít ở chỗ thêm cách nhận biết chỉ cần tái đánh giá với các chính sách cập nhật thôi ạ.

**Phần thứ ba:** cập nhật là thêm chính sách.

Đối với cái này là cái hướng mới mở của em. Giả thiết rằng không tồn tại cả hai chính sách cùng bảo vệ một vùng dữ liệu mà hai giá trị Effect trái ngược nhau.

Ở phần này a giúp em một phần thôi ạ: đối với các chính sách không tìm được policy tương ứng thì chỉ xét trên các policy vừa mới được cập nhật.

**Phần 3:** thiết kế chiến lược chứng minh tính chất định tính của giải thuật

ở phần này chủ yếu xoáy sâu vào việc cập nhật chính sách xảy ra nhiều lần ạ. Cái này khác phần 2 ở chỗ là các chính sách cập nhật sẽ vừa tăng thêm số lượng chính sách và xoá chính sách vừa gán quền (không phải chỉ tồn tại một trường hợp như phần 2).

ở phần này em sẽ có một input và so sánh với output cho trước nếu cả hai cái khớp nhau sẽ hiện chữ pass ạ. Giống như test Unit vậy á anh. Em sẽ nghĩ chiến lược cụ thể khoảng 100 đến 200 test case và sẽ gửi anh sau. Nhưng trong tài liệu này e sẽ gửi cấu trúc cho anh trước ạ.

|  |  |
| --- | --- |
| **Input** | **Output** |
| Chuỗi at và ac |  |
| Câu truy vấn (danh sách giá trị thuộc tính) | Permit / Deny / Not Applicable / Indeterminate |
| Dạng thay đổi bên trong chính sách (1 – Delete (ac hay at muốn delete); 2 – Update(Giá trị hay thuộc tính cần cập nhật); 3 – Insert (ac hay at muốn Insert)) | Trả về câu truy vấn mới  Hoặc giá trị sau khi tái đánh giá chính sách  Hoặc đi đến bước kế tiếp |
| Dạng thay đổi thêm Chính sách | Trả về chính sách được gán cho câu truy vấn |
| Dạng thay đổi xoá chính sách | Trả về chính sách được gán cho câu truy vấn |