$PBAC_B$ Model

Jaehong Park

University of Texas at San Antonio (UTSA), San Antonio, Texas, USA, 78249
jae.park@utsa.edu

$PBAC_B$ Model Definition(1/3)

1. AU, A, AT, O and OR are acting users, action instances, action types, objects, and object roles respectively.
2. G, U, G^{-1} and U^{-1} are sets of role-specific variations of 'wasGeneratedBy' and 'used' dependencies and matching sets of inverse dependencies, respectively.
3. $\{\text{`}c\text{'}, \text{`}c^{-1}\text{'}\}$ is the set of 'wasControlledBy' dependency and its inverse dependency.
4. Base provenance data PD_B forms a directed graph and is formally denoted as a triple $< V_B, E_B, D_B >$:
   - $V_B = AU \cup A \cup O$, a finite set of acting users, action instances, and objects that have been involved in transactions in the system and are represented as vertices;
   - $D_B = \{\text{`}c\text{'}\} \cup U \cup G \cup \{\text{`}c^{-1}\text{'}\} \cup U^{-1} \cup G^{-1}$, a finite set of base dependency types;
   - $E_B \subseteq$ $\{(A \times AU \times \text{`}c\text{'}) \cup (A \times O \times U) \cup (O \times A \times G) \cup (AU \times A \times \text{`}c^{-1}\text{'}) \cup (O \times A \times U^{-1}) \cup (A \times O \times G^{-1})\}$, denoting dependency edges, is the set of existing base dependencies in the provenance data.

$PBAC_B$ Model Definition(2/3)

1. DN_O, disjoint from D_B, is a finite set of abstracted names for dependencies of objects.
2. Let Σ be an alphabet of terms in $D_B \cup DN_O$. The set $DPATH$ of regular expressions is inductively defined as follows:
    - $\forall p \in \Sigma, p \in DPATH; \epsilon \in DPATH;$
    - $(P_1|P_2), (P_1.P_2), P_1*, P_1+, P_1? \in DPATH$, where $P_1 \in DPATH$ and $P_2 \in DPATH$.
3. $DPATH_B \subseteq DPATH$, is the set of regular expression using only alphabet of terms in D_B.
4. $DL_O : DN_O \rightarrow DPATH$, defines each $dn \in DN_O$ as a path expression. DL_O is also viewed as a list of pairs of object dependency names and corresponding dependency paths.
5. $\lambda_O : DN_O \rightarrow DPATH_B$, maps each $dn \in DN_O$ to a path expression using only base dependency types $d_b \in D_B$ by repeatedly expanding the definitions of any $dn_i \in DN_O$ that occurs in $DL_O(dn)$.

$PBAC_B$ Model Definition(3/3)

1. PE is a language specified in the policy expression grammar PG.
2. $P \subseteq PE$, is a finite set of policies.
3. $\gamma : AT \rightarrow P$, a mapping of an action type to a policy.
4. $\delta_O : O \times DPATH_B \rightarrow 2^{V_B}$, a function mapping an object and a base dependency path to vertices in PD_B such that $o_2 \in \delta(o_1, dpath)$ iff there exists a path in PD_B from o_1 to o_2 whose edge labels form a string that satisfies the regular expression $dpath$.

$PBAC_B$ Model Access Evaluation(1/3)

Algorithm 1 $AccessEvaluation(au, a, O)$

```
(Rule Collecting Phase)
$at \leftarrow a$'s action type
$p \leftarrow \gamma(at)$
$RULE_{UA} \leftarrow$ user authorization rules $UARule$ found in $p$
$RULE_{AV} \leftarrow$ action validation rules $AVRule$ found in $p$
(User Authorization Phase)
(Action Validation Phase)
Evaluate a final truth value of $UAuth$ and $AVal$ using conjunctive connective
```

$PBAC_B$ Model Access Evaluation (2/3)

```
(User Authorization Phase)
for all $rules$ in $RULE_{UA}$ do
    Extract the path rule $(ObjRole, DName)$ from $rules$
    Determine the object $o \in O$, whose role is $ObjRole$
    Extract dependency path expression $dpath_b$ in $DPATH_B$ from $DName$ using $\lambda_O$ function
    Determine vertices by tracing base provenance data $PD_B$ through the paths expressed in $dpath_b$ that start from the object $o$ using $\delta_O$ function
    Determine the truth value by evaluating the result against the rule
end for
$UAuth \leftarrow$ a combined truth value based on conjunctive or disjunctive connectives between rules
```

$PBAC_B$ Model Access Evaluation (3/3)

```
(Action Validation Phase)
for all $rules$ in $RULE_{AV}$ do
    Extract path rules $(ObjRole, DName)$ from $rules$
    for all path rules extracted do
        Determine the object $o \in O$, whose role is $ObjRole$
        Extract dependency path expression $dpath_b$ in $DPATH_B$ from $DName$ using $\lambda_O$ function
        Determine vertices by tracing base provenance data $PD_B$ through the paths expressed in $dpath_b$ that start from the object $o$ using $\delta_O$ function
    end for
    Determine the truth value by evaluating the results of all the extracted path rules
end for
$AVal \leftarrow$ a combined result based on conjunctive or disjunctive connectives between rules
```

$PBAC_B$ Model Policy Grammar

$Policy ::= \text{“}allow\text{”} < Req > \text{“} \Rightarrow \text{”} < UARules > \text{“} \wedge \text{”} < AVRules > |$
$\text{“}true\text{”}$

$Req ::= \text{“(”} < ActUser > \text{“,”} < ActType > \text{“,”} < ObjRoles > \text{“)”}$

$ObjRoles ::= < ObjRole > | < ObjRole > \text{“,”} < ObjRoles >$

$UARules ::= < UARule > |\text{“(”} < UARules > \text{“)”}|$
$< UARules >< Connect >< UARules >$

$AVRules ::= < AVRule > |\text{“(”} < AVRules > \text{“)”}|$
$< AVRules >< Connect >< AVRules >$

$Connect ::= \vee|\wedge$

$UARule ::= < ActUser >< oper1 >< PathRule >$

$AVRule ::= \text{“|”} < PathRule > \text{“|”} < oper2 >< Number > |$
$< PathRule >< oper3 >< PathRule >$

$PathRule ::= \text{“(”} < ObjRole > \text{“,”} < DName > \text{“)”}$

$oper1 ::= \text{“} \in \text{”}|\text{“} \notin \text{”}$

$oper2 ::= \text{“} = \text{”}|\text{“} \neq \text{”}|\text{“} \geq \text{”}|\text{“} \leq \text{”}|\text{“} < \text{”}|\text{“} > \text{”}$

$oper3 ::= \text{“} = \text{”}|\text{“} \neq \text{”}|\text{“} \subseteq \text{”}$

$DName ::= dn_1|dn_2|\ldots|dn_n$

$Number ::= [0-9]+$

$ActUser ::= au$

$ActType ::= at_1|at_2|\ldots|at_m$

$ObjRole ::= o_{role_1}|o_{role_2}|\ldots|o_{role_k}$

Q
A