Risk-Aware Role-Based Access Control

Liang Chen **Jason Crampton**

Information Security Group, Royal Holloway, University of London

7th International Workshop on Security and Trust Management

Introduction

- Risk-aware access control was proposed to enable the secure sharing of information within or across multiple organizations
  - An access request is evaluated based on the estimate of the expected costs and benefits of allowing access
  - Risk-aware access control is more permissive than traditional policy-based access control
- Role-based access control (RBAC) has become today's dominant access control paradigm
  - ANSI RBAC standard released in 2004
  - Major IT vendors offer products that support RBAC
- How can we extend role-based access control to become risk-aware?

Motivations

Existing risk-aware RBAC models have limitations

- Existing models have a limited way of access the risk of allowing access requests (only in terms of users' trustworthiness)
- Existing models only support the type of binary decisions, where the accesses with acceptable risk are allowed and denied otherwise
- No existing model considers the incorporation of risk mitigation strategies to support richer types of access control decisions

Outline of talk

- Define new way of looking at RBAC96 authorization semantics
- Risk threshold and risk mitigation
- Risk-aware RBAC models and their ways of computing risk
- Conclusion and future work

RBAC96

- RBAC96 defines a number of basic components: users U, roles R, permissions P, a partially ordered set of roles $RH \subseteq R \times R$, a user-role assignment relation $UA \subseteq U \times R$, and a permission-role assignment relation $PA \subseteq P \times R$
- A graph-based formalism of RBAC96 provides a simple way of evaluating access requests
  - We represent an RBAC96 state as an acyclic directed graph $G = (V, E)$, where $V = U \cup R \cup P$, and $E = UA \cup PA \cup RH$
  - An *authorization path* (*au-path*) between v_1 and v_n is a sequence of vertices $v_1, \ldots, v_n$ such that $(v_i, v_{i+1}) \in E$, $i = 1, \ldots, n-1$
  - A user $u \in V$ is authorized for $p \in V$ if and only if there exists an au-path $u = v_1, \ldots, v_n = p$

Risk threshold and risk mitigation

- We assume the existence of a risk domain $\mathcal{D} = [0, 1]$
  - We write $[t, t')$ to denote the risk interval $\{x \in \mathcal{D} : t \leqslant x < t'\}$
- Given a request (u, p), we write $risk(u, p)$ to denote the risk of allowing u to perform some permission p
- We associate each permission with a *risk mitigation strategy* $[(0, \perp), (t_1, b_1), \ldots, (t_{n-1}, b_{n-1}), (t_n, \perp)]$, where $0 < t_1 < \cdots < t_n \leqslant 1$, $b_i \in B$ is some system obligation, and $\perp$ denotes null obligation
  - The request (u, p) is permitted if $risk(u, p) < t_1$
  - The request (u, p) is permitted with the enforcement of b_i if $risk(u, p) \in [t_i, t_{i+1})$
  - The request (u, p) is denied if $risk(u, p) \geqslant t_n$

Defining the risk of allowing access

Generally, given a request (u, p), $risk(u, p)$ can be determined by the *cost* and *likelihood* of p being misused

- Our approach to the definition of risk mitigation strategies on a per-permission basis suggests that we can ignore the cost of p's misuse when considering the risk of granting p
- There are at least three possible ways of qualifying the likelihood of p being misused
  - The degree of *trustworthiness* of users who request to invoke p
  - The degree of *competence* of a user-role assignment
  - The degree of *appropriateness* of a permission-role assignment
- We develop three simple models for risk-aware RBAC which embody the three distinct ways of computing $risk(u, p)$

RBAC$_T$

RBAC$_T$ augments RBAC96 with risk mitigation strategies on permissions and a function $\alpha : U \to (0, 1]$ which is used to specify users's trustworthiness

- Given a request (u, p), we write $\Pi(u, p)$ to denote the set of au-paths between u and p
- We define a risk function $risk_T : U \times P \to [0, 1]$, where

$$risk_T(u, p) = \begin{cases} 1 - \alpha(u) & \text{if } \Pi(u, p) \neq \emptyset \\ 1 & \text{otherwise} \end{cases}$$

RBAC_C

Unlike RBAC_T, RBAC_C defines a function $\beta : U \times R \to (0, 1]$ which specifies users's degree of competence to perform roles

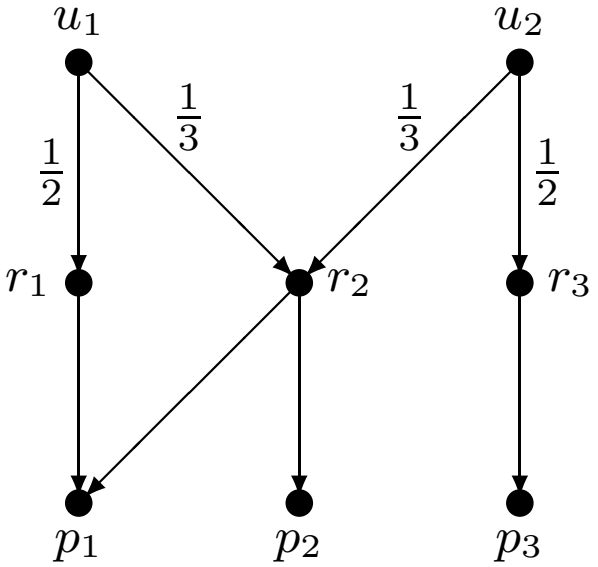
- Informally, given a request (u, p), $risk(u, p)$ is determined by finding a role r for which u is most competent and that lies on an au-path from u to p
- We define a risk function $risk_C : U \times P \to [0, 1]$, where

$$risk_C(u, p) = \begin{cases} 1 & \text{if } u* \cap \uparrow p = \emptyset \\ 1 - \max\{\beta(u, r) : r \in u* \cap \uparrow p\} & \text{otherwise} \end{cases}$$

  - $u*$ is a set of roles for which u is explicitly assigned
  - $\uparrow p$ is a set of entities that are authorized for p

A simple example

- $u_1* = \{r_1, r_2\}$ with $\beta(u_1, r_1) = \frac{1}{2}$ and $\beta(u_1, r_2) = \frac{1}{3}$
- u_1 is able to perform p_1 through the role r_1 for which u_1 is most competent, and hence $risk_C(u_1, p_1) = \frac{1}{2}$
- $risk_C(u_1, p_3) = 1$ as there is no au-path from u_1 to p_3



RBAC_A

RBAC_A introduces a function on permission-role assignments, $\gamma : P \times R \to (0, 1]$ which specifies the degree of appropriateness with which permissions are assigned to roles

- Given a request (u, p), $risk(u, p)$ is determined by a role that u can activate and that is the most appropriate role to which p is assigned
- We define a risk function $risk_A : U \times P \to [0, 1]$, where

$$risk_A(u, p) = \begin{cases} 1 & \text{if } *p \cap \downarrow u = \emptyset \\ 1 - \max\{\gamma(p, r) : r \in *p \cap \downarrow u\} & \text{otherwise} \end{cases}$$

  - $*p$ is a set of roles to which p is explicitly assigned
  - $\downarrow u$ is a set of entities for which u is authorized

A complete model for risk-aware RBAC

We introduce a risk-aware RBAC model that combines the features of the RBAC_T, RBAC_C and RBAC_A models

- Given a request (u, p), $risk(u, p)$ can be computed by finding an au-path between u and p with a minimum risk, but how can we compute the risk associated with each au-path from u to p?
- There are at least two approaches to computing the risk associated with an au-path $u, r, \ldots, r', p$ based on α, β and γ
  - $1 - \min\{\alpha(u), \beta(u, r), \gamma(r', p)\}$
  - $\min\{1, (1 - \alpha(u)) + (1 - \beta(u, r)) + (1 - \gamma(r', p))\}$

Other stuff in the proceedings

- Examine the advantages of flat risk-aware RBAC
- Consider sessions in risk-aware RBAC

Contributions

- We examine three possible ways of defining risk in different components of RBAC96
- We provide a sophisticated treatment of risk mitigation strategies at permission level
- We develop a family of risk-aware RBAC models which differ in the way of measuring and computing risk
- Unlike existing work, our models:
  - have clear authorization semantics
  - support richer types of access control decisions

Current and Future work

- Extend our risk-aware models to include user obligations, and use the idea of "charging for risk" to enforce those obligations
- Construct RBAC_C and RBAC_A states from a given RBAC96 state
  - Investigate a way of defining β values on those user-role assignments which are not encoded in a given RBAC96 state
  - Propose an approach to defining γ values on permission-role assignments based on a given RBAC96 state
- Develop a risk-aware auto-delegation mechanism for RBAC
  - Develop an auto-delegation RBAC model using our risk-aware approach
  - Examine a way of combining risk mitigation with auto-delegation RBAC policies

Questions?