RBAC

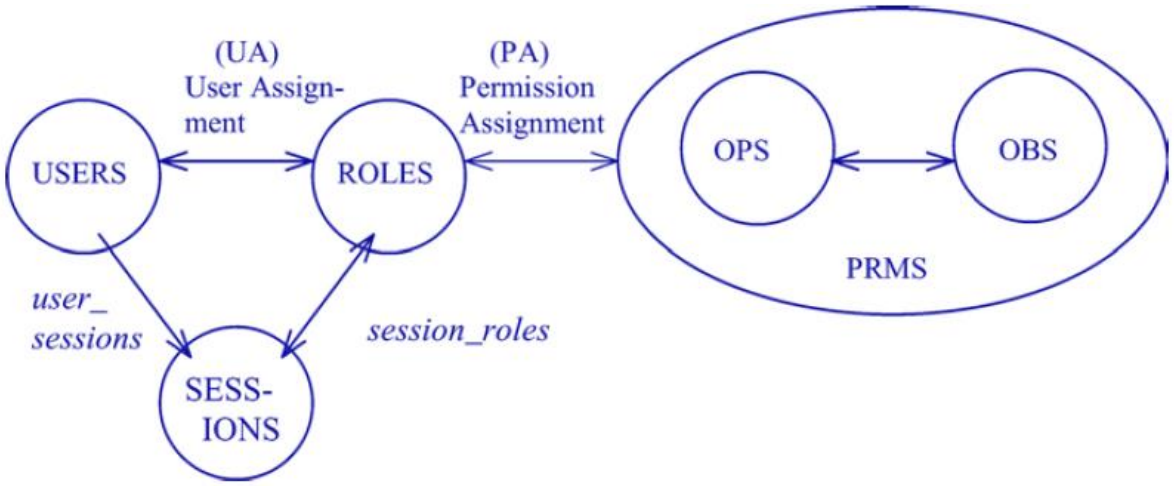
Role Based Access Control

RBAC characteristics

- Least Privilege
  - تخصیص حداقل امتیاز موردنیاز هر نقش
- Separation of Duty
  - تعریف نقش‌های متنافر یا دوبه‌دو ناسازگار
- Abstraction
  - مجوزهای Abstract

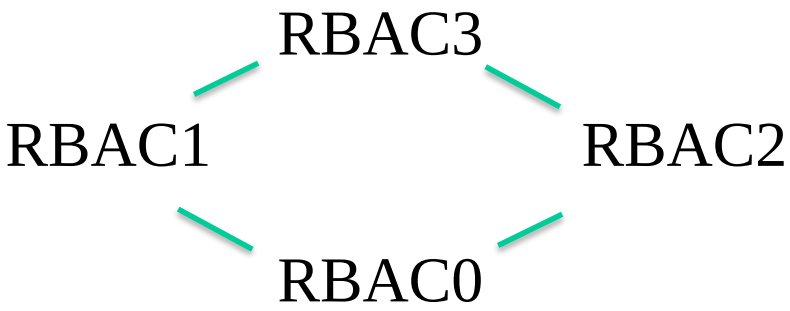
RBAC elements

- Users
- Roles
- Permissions
- Session



RBAC versions

- RBAC0 : core RBAC
- RBAC1 : Hierarchical RBAC
- RBAC2 : constrained RBAC
- RBAC3



Formal Definitions- RBACO

- Basic Sets: U, R, S, Ops, Obs

- $UA \subseteq U * R$

- رابطه انتساب کاربران به نقش‌ها

Formal Definitions- RBAC0

- Assigned users
- نگاشت بین کاربران و نقش‌های منتسب به هر یک از آن‌ها
- $assigned\ users : R \rightarrow \mathcal{P}(U)$
- $\mathcal{P}$: powerset
- $assigned\ users(r) = \{u \in U \mid < u, r > \in UA\}$

Formal Definitions- RBACO

- مجموعه P : مجموعه مجوزهای ممکن
- هر مجوز، مجموعه‌ای از اعمال روی تعدادی اشیاء است.
- $P = \mathcal{P}(\mathrm{Ops} * \mathrm{Obs})$
- نگاشت بین کاربران و نقش‌های منتسب به هر یک از آن‌ها
- $PA \subseteq P * R$
- رابطه انتساب مجوزها به نقش‌ها

Formal Definitions- RBAC0

- Assigned permissions
- نگاشت بین مجوزها و نقش‌ها
- $assigned\ permissions : R \rightarrow \mathcal{P}(P)$
- $\mathcal{P}$: powerset
- $assigned\ permissions(r) = \{p \in P \mid < p, r > \in PA\}$

Formal Definitions- RBACO

- نشست: هر کاربر، نشستی را برقرار و در هر نشست، زیرمجموعه‌ای از نقش‌های خود را فعال می‌کند.
- $user\ sessions$
- نشست‌های مربوط به هر کاربر را مشخص می‌کند.
- $user\ sessions : U \rightarrow \mathcal{P}(S)$

Formal Definitions- RBACO

- $session\ user$
  - کاربر مربوط به هر نشست را مشخص می‌کند.
- $sessions\ user : S \rightarrow U$
- رابطه میان دو تابع $user\ sessions$ و $session\ user$:
  - $session\ user(s) = u \leftrightarrow s \in user\ sessions(u)$

Formal Definitions- RBACO

- $session\ roles$
- در هر نشست، کاربر زیرمجموعه‌ای از نقش‌های منتسب به خود را فعال می‌کند.
- $session\ roles : S \rightarrow \mathcal{P}(R)$
- رابطه بین $session\ roles$ و UA :
- $session\ roles(s) \subseteq \{r \in R \mid < session\ user(s), r > \in UA\}$

Formal Definitions- RBACO

- $available\ session\ permissions$
- در یک نشست، زیرمجموعه‌ای از نقش‌ها فعال شده است؛ مجموعه مجوزهای فعال کاربر در نشست چیست؟
- $available\ session\ permissions(s) = \bigcup_{r_i \in session\ roles\ (s)} assigned\ permissions(r_i)$

RBAC1

- در این مدل، سلسله‌مراتب نقش‌ها به مدل RBAC0 اضافه می‌شود.
- در سلسله‌مراتب نقش‌ها، اگر r_1 زبرنقش r_2 باشد(مثل نقش مدیر که زبرنقش کارمند است)؛
- r_1 علاوه بر مجوزهای خاص خودش که صراحتاً دریافت می‌کند(مجوزهای صریح)، مجوزهای r_2 را نیز به ارث می‌برد(مجوزهای ضمنی).
- همچنین، r_2 علاوه بر کاربران خاص خودش که صراحتاً به آن منتسب می‌شوند(کاربران صریح)، کاربران r_1 را نیز به ارث می‌برد(کاربران ضمنی).

RBAC1

- دو نوع سلسله مراتب:
  - General Role Hierarchy
  - Limited Role Hierarchy

- سلسله مراتب نقش عمومی: سلسله مراتب هر شکلی می‌تواند داشته باشد.
  - ارث بری چندگانه کاربران (ارث بری از بالا به پائین) ممکن است.
- سلسله مراتب نقش محدود: شکل سلسله مراتب درختی است.

Formal Definitions- RBAC1

- $RH \subseteq R * R$
- رابطه با ترتیب جزئی روی R : رابطه زبرنقشی (در برخی منابع: رابطه ارث‌بری) که بصورت $\geq$ نمایش داده می‌شود.
- $r_1 \geq r_2$
- که در آن r_1 زبرنقش و r_2 زیرنقش است.
- $authorized\ permissions(r_2) \subseteq authorized\ permissions(r_1)$
- خروجی تابع $authorized\ permissions$ مجموع مجوزهای صریح و ضمنی است.
- $authorized\ users(r_1) \subseteq authorized\ users(r_2)$
- خروجی تابع $authorized\ users$ مجموع کاربران صریح و ضمنی است.

Formal Definitions- RBAC1

- $authorized\ users : R \rightarrow \mathcal{P}(U)$
- $authorized\ users(r) = \{u \in U \mid \exists r', r' \geq r, < u, r' > \in UA\}$

- $authorized\ permissions : R \rightarrow \mathcal{P}(P)$
- $authorized\ permissions(r) = \{p \in P \mid \exists r', r \geq r', < p, r' > \in PA\}$

RBAC2

- در این مدل، محدودیت‌هایی به مدل RBAC0 اضافه می‌شود.
- مهم‌ترین این محدودیت‌ها، محدودیت‌های تفکیک وظایف (Separation of Duty) یا به‌اختصار SoD است.
- محدودیت‌های تفکیک وظایف از سوءاستفاده افراد به دلیل کسب اختیارات بیش از حد جلوگیری می‌کند.
- به طور مثال در سیستم بانکی، دو نقش صادرکننده چک و تأییدکننده چک، نمی‌توانند به یک فرد منتسب شوند.

RBAC2

- انواع محدودیت‌های SoD :
  - تفکیک وظایف ایستا یا Static SoD یا باختصار SSoD
  - تفکیک وظایف پویا یا Dynamic SoD یا باختصار DSoD

- SSoD : اعمال محدودیت در انتساب نقش به کاربر (روی UA اعمال می‌شود)
- DSoD : اعمال محدودیت حین فعال‌سازی نقش‌ها در هر نشست

Formal Definitions- RBAC2

- SSoD : هیچ کاربری نمی‌تواند n نقش یا بیشتر از یک مجموعه نقش‌های متنافر داشته باشد.
- محدودیت SSoD : $< rs,n >\ ,rs \subseteq R\ ,n \in \mathbb{N}$
- $\mathbb{N}$: مجموعه اعداد طبیعی
- $SSoD\ \subseteq\ \mathcal{P}(R) * \mathbb{N}$
- تأثیر روی UA
- $\forall < rs, n > \in SSoD\ , \forall t \subseteq rs\ , |t| \geq n$
  $\Rightarrow \bigcap_{r \in t} assigned\ users(r) = \phi$

Formal Definitions- RBAC2

- DSoD : در یک نشست n نقش یا بیشتر از مجموعه نقش‌های متنافر نمی‌توانند فعال شوند.
- محدودیت DSoD : $n \in \mathbb{N}, rs \subseteq R, < rs,n >$
- $\mathbb{N}$: مجموعه اعداد طبیعی
- $DSoD \subseteq \mathcal{P}(R) * \mathbb{N}$
- تأثیر روی UA
- $\forall < rs, n > \in DSoD, \forall s \in S \quad s.th \, |session\ roles(s) \cap rs| < n$

RBAC2

- اما در **RBAC2** علاوه بر محدودیت‌های تفکیک وظایف، محدودیت‌های دیگری نیز وجود دارد.

- محدودیت‌های دانه‌بندی یا **Granularity**
  - حداکثر تعداد نقش‌هایی که کاربر می‌تواند اتخاذ کند.
  - حداکثر تعداد کاربرانی که یک نقش می‌تواند داشته باشد.
  - یک مجوز خاص به چند نقش منتسب شود.
  - یک نقش خاص نهایتاً چند مجوز داشته باشد.
  - و ...

RBAC2

- محدودیت‌های نقش‌های پیش‌نیاز یا Pre-requistic Roles
- زمانی کاربر بتواند نقش **A** را اتخاذ کند که قبل از آن نقش **B** را داشته باشد.

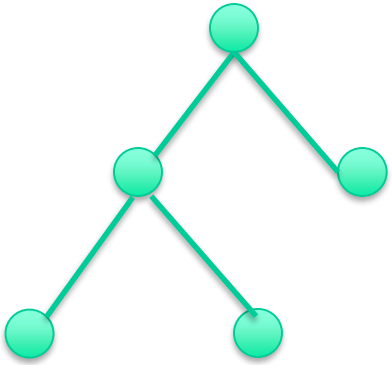
- محدودیت‌های مجوزهای پیش‌نیاز یا Pre-requistic Permissions
- ...

RBAC3

- ترکیب مدل‌های RBAC1 و RBAC2
- برای سادگی بحث فرض می‌کنیم که در محدودیت‌های SoD : $n = 2$
- و بدین ترتیب این محدودیت‌ها را به جای مجموعه و عدد، به صورت دوتایی‌های نقش‌های متنافر نشان می‌دهیم. یعنی به صورت
- $< r_1, r_2 >$ که نشان می‌دهد این دو نقش دوبه‌دو ناسازگار یا متنافر هستند.
- همچنین برای سادگی تنها SSoD را در مدل RBAC3 در نظر می‌گیریم.

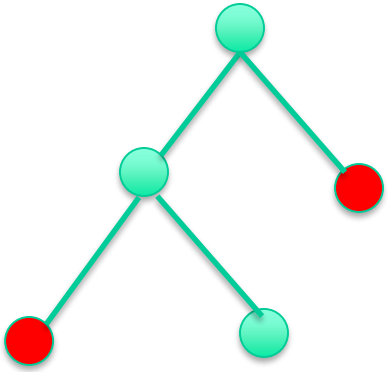
توی این روش هم سلسله مراتب نقش هارو داریم و هم محدودیت ها رو داریم
توی محدودیت ها اصلی ترینش تفکیک وظایف بود
فقط اونجایی که UA داریم می خوایم محدودیت تفکیک وظایف داشته باشیم
برای سادگی اینجا n=2 میگیریم ینی مجموعه رو به صورت دوتایی هایی تعریف میکنیم و هر دوتا نقش متنافر را توی یه دوتایی قرار بدیم
ینی نقش r1 و r2 رو کاربر نمی تونه توامان داشته باشه
پس برای ساده سازی RBAC3 اولا محدودیت SSOD در نظر میگیریم و با فرض این که n=2 است میایم نقش های متنافر رو تعریف می کنیم

RBAC3



- تأثیر SSoD بر سلسله مراتب چیست؟

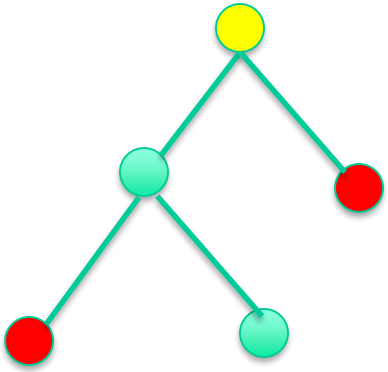
RBAC3



- تأثیر SSoD بر سلسله مراتب چیست؟

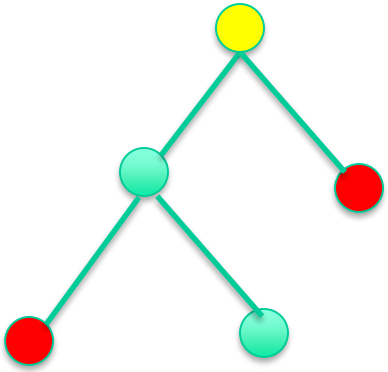
قرمز ها نقش های متنافر هستند توی این حالت سلسله مراتب دیگه نمیتونه به شکل روبرو باشه چون ما یه نقشی داریم به اسم نقش زردرنگ و این زرد زیرنقش دوتا قرمزها هست زرد مجوزهایی دوتا قرمزها رو به ارث می بره و با داشتن مجوزهای هر دو تا قرمز اون امتیاز ویژه رو می تونه به دست بیاره
مثلا یکی از قرمزا میشن صادر کننده چک و یکی دیگه تایید کننده چک و زرد الان این دوتارو داره و خودش همه رو انجام میده
پس ما میخوایم سلسله مراتب دیگه این شکلی نباشه و جوری بچینیم که نقشی نباشه که زبرنقش دوتا نقش باشه
و به صورت فرمال میشه ص 31

RBAC3



- تأثیر SSoD بر سلسله مراتب چیست؟

RBAC3



- تأثیر SSoD بر سلسله مراتب چیست؟

- $\forall < r, r' > \in SSoD \Rightarrow \nexists r" , r" \geq r \ \wedge \ r" \geq r'$

توی تعریف:
این شرطو روی سلسله مراتب می ذاریم
ینی یه نقشی مثل r زگن وجود نداشته باشه که هم زبر نقش r باشه و هم r پرین

RBAC3

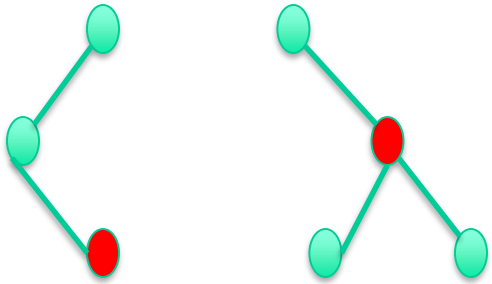
- اعمال این شرایط در UA چگونه است؟

- $\forall < r, r' > \in SSoD \Rightarrow \nexists u \in U , \nexists r_1, r_2 \in R \; s.th \; r_1 \geq r \; \wedge r_2 \geq r' \; \wedge < u, r_1 > \in UA \; \wedge < u, r_2 > \in UA$

RBAC3

- اعمال این شرایط در UA چگونه است؟

- $\forall < r, r' > \in SSoD \Rightarrow \nexists u \in U , \nexists r_1, r_2 \in R \; s.th \; r_1 \geq r \; \wedge r_2 \geq r' \; \wedge < u, r_1 > \in UA \; \wedge < u, r_2 > \in UA$

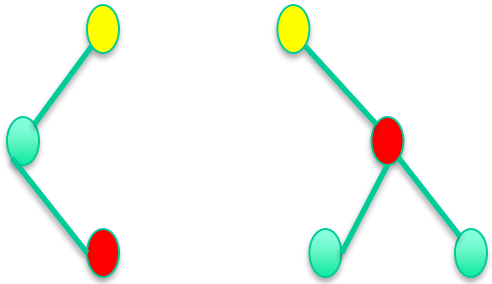


این یه مثال دیگه است
زبرنقش r و r پرین نباید همزمان به یک کاربر برسه
ما نمیخوایم کاربر ما همزمان هم r1 نسبت داده شده باشه توی رابطه UA و هم r2
چه اتفاقی میافته اگر همزمان نسبت داده باشیم؟

RBAC3

- اعمال این شرایط در UA چگونه است؟

- $\forall < r, r' > \in SSoD \Rightarrow \nexists u \in U, \nexists r_1, r_2 \in R \; s.th \; r_1 \geq r \wedge r_2 \geq r' \wedge < u, r_1 > \in UA \wedge < u, r_2 > \in UA$



اگر کاربر هم نقش r1 داشته باشه و هم نقش r2 کاربر ما مجوزهای دوتا قرمزا رو باز داره
اگر دوتا قرمز ها ینی r و r پرین دوتا نقش متنافر باشند دوتا زرد ها هم با هم متنافر خواهند بود بخاطر فرزند های خودشون که متنافر هستند

RBAC3

- روش دیگر بیان این شرط

- $\forall < r, r' > \in SSoD \Rightarrow authorized\ users(r) \cap authorized\ users(r') = \phi$