به نام خدا

1)

DAC:

- مزایا:
  - افزایش انعطاف‌پذیری برای کاربران در تعیین دسترسی به منابع
  - اجازه به کاربران برای تعیین دقیق دسترسی‌ها به فایل‌ها و منابع
- معایب:
  - امکان ایجاد ضعف امنیتی در مواردی که کاربران تصمیم‌گیرنده اصلی هستند

MAC:

- مزایا:
  - ارتقاء امنیت با اعمال سیاست‌های اجباری توسط سیستم عامل
  - کنترل دقیق بر دسترسی به منابع با توجه به تصمیم‌های سازمانی
- معایب:
  - محدودیت‌های انعطاف‌پذیری برای کاربران و ادمین‌ها

RBAC:

- مزایا:
  - مدیریت آسان‌تر و بهبود امنیت با تخصیص دسترسی‌ها بر اساس نقش‌ها
  - کاهش اشکالات مرتبط با مدیریت دسترسی در مقیاس بزرگ
- معایب:
  - پیچیدگی افزایش یافته در اجرای سیستم‌های با ساختار پیچیده

ABAC:

- مزایا:
  - افزایش امنیت با ارتباط دسترسی‌ها به ویژگی‌های افراد و منابع
  - تصمیم‌گیری دقیق‌تر بر اساس ویژگی‌های متغیر
- معایب:
  - پیچیدگی بیشتر در پیاده‌سازی و مدیریت نسبت به سایر روش‌ها

2)

DAC:

- سیستم‌های عامل ویندوز و لینوکس برای اعطای دسترسی‌های مختلف به کاربرها

MAC:

- سیستم‌های امنیتی دولتی و نظامی که از انواع سیستم‌های امنیتی چند لایه هستند(multi-level security system)
- SELinux (ارائه شده توسط Red Hat Software و Secure Computing Corporation) که یک ماژول امنیتی کرنل لینوکس بوده و شامل یک سری تغییرات در کرنل لینوکس از جمله اضافه کردن MAC می‌باشد که در برخی از توزیع های لینوکس (Fedora, Debian, Ubuntu) گنجانده شده است.
- سیستم عامل اندروید (بر پایه SELinux) منبع

RBAC:

- نرم افزارهایی که هر کدام به نحوی نقش‌های سازمانی را بازنمایی می‌کنند از قبیل سیستم های مدیریت محتوا(wordpress)، مدیریت پروژه(Trello, Jira، میزیتو)، مدیریت منابع انسانی، مدیریت مالی و حسابداری، مدیریت ارتباط با مشتری که در آنها هر یک از کاربران یک نقش یا سمت مشخص (نویسنده، ویرایشگر، مدیر پروژه، توسعه دهنده، مدیر، حسابدار، منشی، صندوق‌دار، مشتری و ...) داشته و بر اساس آن نقش دسترسی‌ها به او اعطا می‌گردد.

ABAC:

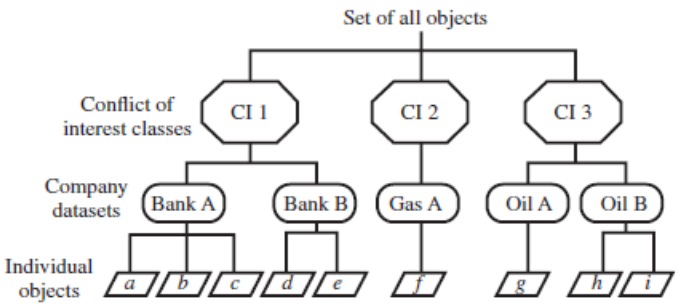
- BeyondCorp: این یک مدل امنیتی در سطح سازمان است که پیاده سازی مدل اعتماد صفر (zero trust model) توسط گوگل می‌باشد و از ABAC استفاده می کند. دسترسی به منابع بر اساس ویژگی های کاربر و منبع تعیین می شود. در این مدل، همه کاربران در ابتدا به عنوان ناامن در نظر گرفته می شوند و باید احراز هویت شده و مجوزهای لازم را برای دسترسی به منابع دریافت کنند. این به شرکت گوگل اجازه می دهد که کنترل دقیق تری بر دسترسی کاربران داشته باشد و از امنیت ارتباط خود با کاربران در بستر یک شبکه ناامن اطمینان داشته باشد.

3)

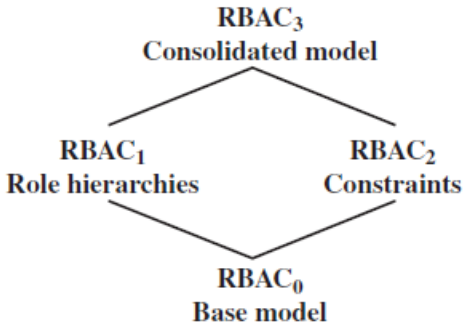
Clark-Wilson Integrity Model: این مدل تمرکز اصلی خود را بر روی یکپارچگی داده‌ها قرار می‌دهد. اصل این مدل بر پایه تعریف «تراکنش‌های خوش فرم» (well-formed transactions) و «تفکیک وظایف کاربران» (separation of duties among users) است. تراکنش‌های خوش فرم تغییراتی هستند که سیستم را از یک حالت امن به حالت امن دیگر می‌برند. این تغییرات باید توسط برنامه‌هایی اعمال شوند که توسط مدیر سیستم تایید شده‌اند. تفکیک وظایف به این معناست که فردی که اجازه اجرای یک تراکنش را می‌دهد، نباید قادر به اجرای همان تراکنش باشد. این اصول به حفاظت از اطلاعات در برابر تغییرات غیرمجاز و خطاهای ناخواسته کمک می‌کنند.

Biba Integrity Model: این مدل همچنین تمرکز خود را بر روی حفظ یکپارچگی داده‌ها قرار داده است. این مدل بر پایه دو اصل اصلی است. اصل اول «ممنوعیت خواندن از پایین» (no read down) نام دارد که به معنای این است که یک عامل در یک سطح یکپارچگی بالا نمی‌تواند یک شئ در یک سطح یکپارچگی پایین‌تر را بخواند. این برای جلوگیری از آلوده شدن عامل توسط داده‌های با قابلیت اعتماد کمتر است. اصل دوم «ممنوعیت نوشتن در بالا» (no write up) نام دارد که به معنای این است که یک عامل در یک سطح یکپارچگی پایین نمی‌تواند در یک شئ در یک سطح یکپارچگی بالاتر چیزی بنویسد. این اصل به منظور جلوگیری از خرابی داده‌های با قابلیت اطمینان بیشتر توسط عامل است.

مدل دیوار چینی: این مدل، برای جلوگیری از تداخل منافع با محدود کردن دسترسی به اطلاعات طراحی شده است. این مدل با طبقه‌بندی اشیاء به کلاس‌های تداخل منافع بر اساس حساسیت اطلاعاتی که درون آنها قرار دارد کار می‌کند. این مدل از دسترسی عامل به یک شئ در صورت ایجاد بستر تعارض منافع جلوگیری می‌کند. این مدل دو قانون اصلی دارد. قانون اول: هر عامل فقط زمانی می‌تواند یک شئ را بخواند که یا قبلاً از مجموعه‌ای که این شئ در آن وجود دارد خوانده باشد و یا تا به حال از هیچ کلاس تعارض منافع دیگری چیزی نخوانده باشد. قانون دوم: هر عامل صرفاً زمانی می‌تواند در یک شئ چیزی بنویسد که اولاً اجازه خواندن آن شئ را طبق قانون اول داشته باشد، ثانیاً تمامی اشیائی که عامل اجازه خواند از آنها را دارد، باید در مجموعه داده های یکسان با این شئ باشند.



4)



$RBAC_0$: مدل پایه شامل مفاهیم کاربر، نقش، دسترسی، جلسه که یک سری روابط چند به چند بین کاربران و نقش‌ها، نقش‌ها و دسترسی‌ها وجود دارد. همچنین روابط یک به چند بین هر کاربر و یک مجموعه از نقش‌ها در قالب مفهوم جلسه وجود دارد.

$RBAC_1$: همان مدل پایه به همراه مفهوم سلسله مراتب؛ به این معنا که نقش‌ها می‌توانند به گونه‌ای تعریف شوند که یک نقش، تمامی دسترسی‌های نقش‌های زیر مجموعه خود را در بر گرفته و انتساب یک نقش به معنای انتساب تمامی نقش‌های زیرمجموعه آن به کاربر است.

$RBAC_2$: مدل پایه به اضافه یک سری محدودیت‌ها؛ که شامل محدودیت‌های انحصار متقابل، محدودیت‌های عددی و محدودیت‌های پیش‌نیازی است.
محدودیت‌های انحصار متقابل مربوط به شرایطی است که این ویژگی‌ها را دارند: الف) یک کاربر تنها می‌تواند به یک نقش در یک مجموعه منتسب شود. ب) یک دسترسی صرفاً می‌تواند به یک نقش اعطا شود.

محدودیت‌های عددی به معنای ایجاد سقف برای تعداد کاربرانی که یک نقش را دارند، تعداد نقش‌هایی که به یک کاربر انتساب می‌یابد یا تعداد نقش‌هایی که در یک جلسه به یک کاربر منتسب می‌شود است.

محدودیت‌های پیش‌نیازی نیز به این معناست که به طور مثال انتساب یک نقش به یک کاربر مشروط به آن است که قبل از آن، آن کاربر یک نقش دیگر را داشته باشد.

$RBAC_3$: این مدل نیز ترکیب دو مدل $RBAC_1$ و $RBAC_2$ است.