امنیت در

پایگاه داده‌های کامپیوتری

۱ - مقدمه

در طی سه دهۀ اخیر تعداد پایگاه داده‌های کامپیوتری افزایش بسیاری داشته است. حضور اینترنت به همراه توانائیهای شبکه، دسترسی به داده و اطلاعات را آسانتر کرده است. به عنوان مثال، کاربران امروزه می‌توانند به حجم بالایی از اطلاعات در فاصلۀ زمانی بسیار کوتاهی دسترسی پیدا کنند. به همین نسبتی که ابزارها و تکنولوژی دسترسی و استفاده از اطلاعات توسعه می‌یابند، نیاز به حفاظت اطلاعات هم بوجود می‌آید. بسیاری دولتها و سازمانها صنعتی داده‌های مهم و طبقه بندی شده‌ای دارند که باید حفاظت شوند. سازمانهای بسیار دیگری هم مثل مؤسسات دانشگاهی نیز اطلاعات مهمی در مورد دانشجویان و کارمندانشان دارند. در نتیجه تکنیکهایی برای حفاظت داده های ذخیره شده در سیستمهای مدیریت پایگاه داده،[1] اولویت بالایی پیدا کرده‌اند.

در طول سه دهۀ اخیر، پیشرفتهای بسیاری در مورد امنیت پایگاه داده‌ها حاصل شده است. بسیاری از کارهای اولیه، روی امنیت پایگاه داده‌های آماری انجام شد. در دهۀ ۷۰، همزمان با شروع تحقیقات روی پایگاه داده‌های رابطه‌ای، توجه مستقیماً به مسئله کنترل دسترسی[2] بود و بیشتر از همه، کار روی مدلهای کنترل دسترسی احتیاطی[3] شروع شد. در حالی که، در سالهای پایانی دهۀ ۷۰، کار بروی امنیت الزامی[4]

---

[1]- DBMS
[2]- Access Control
[3]- Discretionary
[4]- Mandatory

ولی در واقع تا مطالعات نیروی هوایی در ۱۹۸۲، که تلاش وسیعی برای DBMSهای امن چند سطحی[1] بود، کار مهمی انجام نشد.

در هزارهٔ جدید با حضور تکنولوژیهایی مثل کتابخانه‌های دیجیتال، شبکه گسترده جهانی و سیستمهای محاسباتی اشتراکی، علاقه بسیاری به امنیت نه تنها در بین سازمانهای دولتی، بلکه بین سازمانهای اقتصادی هم وجود دارد. این مقاله مروری به پیشرفتها و محصولات در سیستمهای پایگاه داده‌ای امن در دو زمینهٔ اجباری و احتیاطی دارد.

۲ کنترل دسترسی (مفاهیم و سیاستها)

دراین بخش مفاهیم پایه در کنترل دسترسی معرفی می‌شوند. سپس در مورد سیاستهای کنترل دسترسی احتیاطی و الزامی بحث می‌شود و نهایتاً مروری داریم بر سیاستهای سرپرستی.

[1]- Multilevel

۲ـ۱ مفاهیم اساسی

کنترل دسترسی معمولاً در مقابل مجموعه ای از قوانین اعطای مجوز که توسط مدیران امنیتی یا کاربران براساس بعضی سیاستهای خاص ارائه می‌شوند، قرار دارد.

قانون اعطای مجوز، در حالت کلی بیان می‌کند که فرد[۱] S اجازه دارد که امتیاز[۲] P را بروی شیئی[۳] O بکار ببرد.

اشیاء مجاز[۴] : ترکیبات غیرفعال سیستم هستند که باید در مقابل دسترسی‌های غیرمجاز محافظت شوند. اشیایی که باید به آنها متوجه شدند به مدل داده‌ای مورد استفاده بستگی دارند. به عنوان مثال، در یک سیستم عامل فایلها و دایرکتوریها اشیاء هستند. در حالیکه، در یک DBMS منابعی که باید محافظت شوند رابطه ها، دیدها و صفات هستند.

اشخاص مجاز[۵] : موجودیتهایی در سیستم هستند که اجازهٔ دسترسی به آنها داده می‌شود. اشخاص به دسته‌های زیر تقسیم بندی می‌شدند :

- کاربران : که شخصیتهای مجزا و مشخصی هستند که با سیستم در ارتباطند.
- گروهها مجموعه ای از کاربران.
- نقشها[۱] : مجموعه‌ای نامدار از امتیازها که احتیاج دارند، فعالیت خاصی را در رابطه با سیستم انجام دهند.

---

[1]- Subject
[2]-Privilege
[3]- Object
[4]- Authorization Object
[5]- Authorization Subject

- سلسله عملیات[۲] : که برنامه‌هایی را برای کاربر اجراء می‌کند. به طور کلی، سلسله عملیات به آدرسهای حافظه، استفاده از CPU ، فراخوانی برنامه‌های دیگر و عملیات بروی داده اشاره می‌کند.

امتیازهای مجاز[۳] : انواع عملیاتی را که یک فرد می‌تواند روی یک شیئی در سیستم اجراء کند، بیان می‌کند. مجموعهٔ این امتیازها به منابعی که باید محافظت شوند، بستگی دارد. به عنوان مثال، در یک سیستم عامل خواندن، نوشتن و اجراء از امتیازها هستند. ولی، دریک DBMS رابطه‌ای، انتخاب، درج، تغییر و حذف از جمله امتیازها به شمار می‌روند.

۲ـ۲ سیاستهای کنترل دسترسی :

سیاستهای کنترل دسترسی، معیارهایی هستند که براساس آنها تعیین می‌شود آیا یک درخواست دسترسی باید مجاز شمرده شود یا نه. یک طبقه بندی کلی بین سیاستهای کنترل دسترسی احتیاطی و الزامی است.

---

[1]- Role
[2]- process
[3]- Authorization privilege

۲ـ۲ـ۱ سیاست کنترل دسترسی احتیاطی:

سیاستهای کنترل دسترسی احتیاطی (DAC) ، دسترسی افراد به اشیاء را براساس شناسه افراد، قوانین و مجوزها کنترل می‌کند. قوانین برای هر فرد، مجوزهایی را که می‌تواند برای انجام عملیات روی اشیاء بکار برد، بیان می‌کند. وقتی تقاضای درخواستی به سیستم می‌آید، مکانیسم دسترسی مشخصی می‌کند آیا قانونی برای تأیید این درخواست وجود دارد یا نه. اگر قانونی وجود داشت درخواست مجاز شمرده می‌شود، در غیراین صورت رد می‌شود. چنین مکانیزمی احتیاطی است و در آن به اشخاص اجازه داده می‌شود که مجوز دسترسی به داده‌هایشان را به دیگران بدهند.

سیاستهای دسترسی احتیاطی انعطاف پذیری زیادی دارند. به طوری که، اجازه تعریف محدوده وسیعی از قوانین کنترل دسترسی را با استفاده از انواع مختلف مجوزها را می‌دهند. مثل مجوزهای مثبت و منفی و مجوزهای قوی و ضعیف. در زیر توضیح مختصری از هر کدام ارائه می‌شود.

- ***مجوزهای مثبت و منفی:***

در سیستمی که مجوز مثبت دارد. هرگاه فردی بخواهد به شیئی خاصی دسترسی داشته باشد، سیستم چک می‌کند آیا مجوزی وجود دارد و فقط در صورت وجود، به شخص اجازه دسترسی داده می‌شود. عدم وجود مجوز به معنی رد درخواست است. مشکل این خط مشی این است که، عدم وجود مجوز به معنای جلوگیری از

دسترسی شخص به شیئی در آینده نیست. این مشکل توسط مجوزهای منفی حل شد که به معنی رد قطعی مجوز در چنین مواردی است.

- ***مجوزهای قوی و ضعیف :***

بعضی مدلهایی که هر دو مجوز مثبت و منفی را دارند به دو دسته مجوزهای قوی و ضعیف نیز تقسیم می‌شوند. مجوزهای قوی (چه مثبت و چه منفی) باطل نمی‌شوند. در حالیکه، مجوزهای ضعیف براساس قوانین خاصی توسط مجوزهای قوی یا ضعیف دیگری می‌توانند باطل شوند.

۲ـ۲ـ۲ سیاست کنترل دسترسی الزامی

سیاستهای کنترل دسترسی الزامی (MAC) بیان کننده دسترسی است که افراد به اشیاء براساس رده بندی شیئی و فرد دارند. این نوع از امنیت تحت عنوان امنیت چند لایه[1] هم نام برده می‌شود. سیستمهای پایگاه داده‌ای که خصوصیات امنیت چند لایه را تأمین می‌کنند، DBMS های امن چند لایه (MLS/DBMS) یا DBMS های مطمئن نامیده می‌شوند. بیشتر MLS/DBMSها براساس سیاست Bell و Lapadula ، طراحی و ساخته شده‌اند. دراین خط‌مشی‌ها، افراد به عنوان سطوح مجاز مطرح می‌شوند و می‌توانند در سطح مجاز خود عمل کنند. اشیاء به سطوح حساسیت ارجاع می‌شوند. سطوح مجاز حساسیت را سطوح امنیت می‌نامند. آنچه در زیر می‌آید، دو قانون مهم این خط مشی است :

---

[1]- Multilevel Security

ـ *ویژگی امنیتی ساده* : یک فرد دسترسی خواندن یک شیئی را دارد اگر سطح امنیتی آن بر سطح امنیتی شیئی مسلط باشد.

ـ *ویژگی ستاره* : یک فرد دسترسی نوشتن یک شیئی را دارد اگر سطح امنیتی شیئی توسط سطح امنیتی فرد پوشانده شود.

شکل زیر تفاوت بین سیاستهای الزامی و احتیاطی را نشان می‌دهد. تحت سیاست احتیاطی یک تقاضای دسترسی مجاز شمرده می‌شود اگر قانونی وجود داشته باشد که دسترسی را مجاز بداند. در مقابل، در سیاست الزامی یک دسترسی مجاز است، اگر رابطه خاصی بین سطح امنیتی شخصی که تقاضای دسترسی دارد و سطح امنیتی شیئی که مورد تقاضاست، وجود داشته باشد.

۲ـ۳ سیاستهای سرپرستی[1]

یکی دیگر از ابعادی که می تواند معیاری برای مقایسه مدلهای کنترل دسترسی باشد، سیاستهای سرپرستی است، که حمایت می‌کند. سرپرستی به عملیات اعطا و بازپس گرفتن مجوز اطلاق می‌شود. ما سیاستهای سرپرستی را به صورت زیر طبقه بندی می‌کنیم.

سرپرستی *DBA* : تحت این سیاست، فقط DBA می‌تواند حق دسترسی بدهد یا تقاضایی را برگرداند. این سیاست بسیار متمرکز است و امروزه به ندرت در DBMSها بکار می‌رود، مگر در ساده ترین آنها.

[1]- Administration

سرپرستی شیئی - مالک[1] : براساس این سیاست که عمدتاً توسط DBMSها و سیستم عاملها استفاده می‌شود، بوجود آوردند شیئی مالک آن محسوب می‌شود و تنها شخص مجاز برای سرپرستی شیئی است.

سرپرستی متصدی شیئی[2] : بر طبق این سیاست، یک شخص، نه الزاماً ایجاد کننده شیئی، مدیر سرپرستی شیئی است. براساس این سیاست حتی ایجاد کنندهٔ شیئی هم باید مجوز دسترسی به شیئی را دریافت کند.

دومین و سومین سیاست می‌توانند با وکالت سرپرستی و انتقال سرپرستی ترکیب شوند. وکالت سرپرستی به این معناست که مدیر یا سرپرست یک شیئی می‌تواند اعمال سرپرستی بروی یک شیئی را به شخص دیگری واگذار کند. بیشتر DBMSها سیاست سرپرستی براساس سرپرستی مالک با امکان واگذاری را حمایت می‌کنند. باید توجه داشت که تحت خط مشی واگذاری، سرپرست اولیه شیئی امتیاز سرپرستی خود را از دست نمی دهد.

انتقال سرپرستی مثل واگذاری، سرپرستی را به شخص دیگری می‌دهد. با این تفاوت که سرپرست اولیه امتیاز سرپرستی خود را از دست می‌دهد. برای انتقال سرپرستی دو خط مشی زیر وجود دارد.

ـ *ارجاع بازگشتی*[1] : تمام مجوزهایی که توسط سرپرستی پیشین داده شده، به صورت بازگشتی ارجاع داده می‌شود.

---

[1]- Object-Owner
[2]- Object Curator

ـ *انتقال واگذار کننده*[۲] : تمام مجوزهای که توسط سرپرستی پیشین صادر شده نگه داشته می‌شوند.

علاوه براین انتقال می‌تواند با پذیرش[۳] یا بدون پذیرش باشد. پذیرش به معنای این است که شخصی که سرپرستی به او واگذار می‌شود باید صریحاً این مسؤولیت را بپذیرد. انتقال بدون پذیرش به معنای این است که چنین پذیرشی احتیاج نیست.

۳- سیستمها و مدلهای کنترل دسترسی احتیاطی

دراین بخش به بحث در مورد مدلها و سیستمهای DAC می‌پردازیم. مدلهای احتیاطی براساس معیارهای گوناگونی می‌تواند طبقه بندی شود. این بخش این مدلها را براساس DBMSهایی که تحت آن این مدلها توسعه می‌یابند به سه گروه تقسیم بندی می‌کند : مدلهای اعطای مجوز برای DBMSهای رابطه‌ای، مدلهای اعطای مجوز برای DBMSهای شیئی گرا و مدلهای اعطای مجوز برای DBMSهای فعال.

۳ـ۱ مدلهای اعطای مجوز برای DBMSهای رابطه‌ای

دراین بخش مروری داریم بر مدلهای اعطای مجوز که برای DBMSهای رابطه‌ای ساخته شده‌اند و با شرح مدل System R شروع می‌کنیم. مدل System R یک حادثه مهم در تاریخ مدلهای اعطای مجوز است. اهمیت مدل سیستمهای R از آنجایی است که بسیاری DBMSهای تجاری مکانیزم اعطای مجوز را براساس آن توسعه دادند.

دراین مدل اشیایی که باید محافظت شوند جدولها و دیدهایی هستند که اشخاص، امتیازهای گوناگون نسبت به آنها دارند. امتیازهایی که این مدل حمایت می‌کند شامل،

---

[1]- Recursive Revoke
[2]- Grantor Tronsfer
[3]- acceptance

انتخاب برای انتخاب تاپلها از جدول، به روز رسانی برای تغییر تاپلهای یک جدول، درج و حذف برای افزودن یا حذف کردن تاپلهای جدول، حذف جدول برای پاک کردن کل یک جدول. گروه و نقش دراین مدل حمایت نمی‌شوند. این مدل امکانات سرپرستی نامتمرکز را حمایت می‌کند. هرگاه شخصی جدولی را بوجود می‌آورد، امتیازی را نسبت به آن بدست می‌آورد. مالک جدول می‌تواند تمام امتیازها را بر جدول اعمال کند.

این مدل ارجاع بازگشتی دارد، به این معنا که وقتی شخصی مجوز جدولی را از کاربر دیگری می‌گیرد. تمام مجوزهایی که قبلاً به او داده شده ارجاع می‌شود.

۳ـ۲ مدلهای اعطای مجوز برای DBMSهای شیئی گرا

امروزه DBMSهای شیئی‌گرا و شیئی – رابطه‌ای از مهمترین زمینه های تحقیق در حوزه DB هستند. دلیل این اهمیت این است که آنها بسیار مناسب برای کاربردهای پیشرفته مثل CAD/CAM، مولتی مدیا و کاربردهای نقشه‌کشی هستند. چون این برنامه‌ها احتیاج به مدلهای داده‌ای غنی‌تری نسبت به مدلهای رابطه‌ای دارند. احتیاجات سیستمهای DBMSها هم متفاوت از سیستمهای رابطه ای است و این باعث می‌شود مدلهای سنتی برای DBMSهای رابطه‌ای، برای سیستمهای شیئی‌گرا کافی نباشد. با وجود رشد علاقه و توجه به ODBMSها، تحقیقات برای مدلهای اعطای مجوز برای ODBMSها هنوز در مراحل اولیه است. اگرچه طرحهای بسیاری وجود دارد. تنها مدلهای Orion و Iris مدلهای قابل مقایسه با مدلهای RDBMSها دارند.

۳ـ۲ـ۱ مدل Orion

مدل اعطای مجوز Orion، مجوزهای مثبت و منفی و همچنین قوی و ضعیف را حمایت می‌کند. مجوز قوی همیشه اولویت بیشتری نسبت به مجوز ضعیف دارد. مجوزها به جای کاربران تکی به نقشها داده می‌شوند و یک کاربر مجاز است عملی را

روی یک شیئی انجام دهد، اگر نقشی وجود داشته باشد که اجازه این کار را داشته باشد. نقشها، اشیاء و امتیازها تحت یک سلسله مراتب سازماندهی می‌شوند و یکسری قوانین انتشار یا تکثیر اعمال می‌شود :

- اگر یک نقش مجوز دسترسی به یک شیئی را داشته باشد، تمام نقشهای بالاتر از آن در سلسه مراتب همان مجوز را دارند.
- اگر یک نقش مجوز منفی برای دسترسی به یک شیئی را داشته باشد، تمام نقشهای بعداز آن همان مجوز منفی را خواهند داشت.

قوانین انتشار یکسانی برای امتیازها هم تعریف می‌شوند. نهایتاً قوانین انتشار بر یک شیئی اجازهٔ مجوزهایی را می‌دهد که از مجوزهای شیئی که، منقطاً با آن در ارتباط است، مشتق شده باشد. به عنوان مثال مجوز خواندن یک کلاس، مجوز خواندن تمام نمونه‌های آن را صادر می‌کند.

۳ـ۲ـ۲ مدل *Iris*

در مدل Iris ، صفات و متددها هر دو به عنوان تابع تعریف می‌شدند و تنها امتیازی که توسط مدل حمایت می‌شود، فراخوانی تابع است. یک فرد که امتیاز فراخوانی یک تابع را دارد مجاز برای فراخواندن آن تابع است. فردی که ایجاد کننده یک تابع است، مالک آن محسوب می‌شود و به طور خودکار امتیاز فراخوانی آن را دریافت می‌کند. علاوه براین مالک یک تابع می‌تواند امتیاز فراخوانی تابع را به افراد دیگر هم بدهد. این اعطای امتیاز می‌تواند شامل شرایط هم باشد، که به شخصی که امتیاز را می‌گیرد اجازه می‌دهد که آن را به دیگران هم بدهد.

این مدل اجازه می‌دهد که یک امتیاز هم به گروه و هم به کاربر داده شود یا گرفته شود. یک کاربر می‌تواند متعلق به چندین گروه باشد. توابع مشتق شده تحت عنوان

توابع دیگر تعریف می‌شوند. علاوه براین، گروه‌ها می‌توانند تو در تو باشند. مدل Iris دو خط مشی برای حفاظت از توابع مشتق شده دارد.

تحت خط مشی که *مجوز استاتیک* نامیده می‌شود، فردی که تقاضای اجرای یک تابع مشتق را دارد. فقط باید اجازه فراخوانی تابع مشتق را داشته باشد. در خط مشی دیگر، *مجوز دینامیک*، فرا خواننده هم باید مجوز فراخوانی تابع مشتق را داشته باشد و هم مجوز برای فراخوانی تمام توابعی که تابع مشتق آنها را اجراء می‌کند. هنگام ایجاد یک تابع مشتق باید مشخص کنیم که از کدامیک از این دو خط مشی برای بررسی تقاضاهای اجراء استفاده کنیم.

این مدل همچنین دو مفهوم برای کنترل دسترسی تعریف می‌کند: توابع نگهبان[۱] و توابع پراکسی[۲].

توابع نگهبان، ابزاری برای گذاشتن پیش شرط در فراخوانی یک تابع و در نتیجه محدود کردن دسترسی به توابع هستند. تابعی که تابع نگهبان به آن اشاره می‌کند تابع هدف نامیده می‌شود. یک تابع هدف اجراء می‌شود، اگر تابع نگهبان مربوط به آن موفق ارزیابی شود. مهمترین مزیت تابع نگهبان این است که دسترسی به یک تابع محدود می‌شود، بدون اینکه لازم باشد کد تابع تغییر کند. توابع پراکسی، پیاده سازی‌های مختلفی از یک تابع مشخص را برای افراد مختلف، فرد یا گروه، تأمین می‌کند. وقتی یک تابع مورد درخواست واقع می‌شود، تابع پراکسی مناسب به جای تابع اصلی اجراء می‌شود.

۳ـ۳ مدلهای اعطای مجوز برای DBMSهای فعال

---

[1]- guard
[2]- proxy

پایگاه داده‌های فعال با یک سیستم قانونمند که DBMS را قادر می‌سازد با تریگر کردن قانونها نسبت به حوادث عکس‌العمل نشان دهد، تعریف می‌شوند. قوانین اعمالی را توصیف می‌کند که می‌خواهیم به صورت خودکار در هنگام رخ دادن حادثه خاصی یا ارضاء شدن شرط خاصی در DB اجراء شوند. به عنوان یک مثال از این مدل در ادامه امکانات سیستم Starbust را شرح می‌دهیم.

Starbust یک نمونه از سیستم پایگاه داده‌ای رابطه‌ای توسعه پذیر است که در مرکز تحقیقاتی Almaden در IBM تولید شده است. Starbust بوسیلهٔ یک زبان قانونمند توصیف می‌شود. مدل اعطای مجوز آن سلسله مراتبی است و از انواع امتیازها که در DB می‌تواند اعمال شود، حمایت می‌کند. دراین سلسله مراتب عناصر بالاتر، انواع پائین‌تر را پوشش می‌دهند. مثالی از انواع امتیازها، کنترل است که تمام امتیازهای دیگر یعنی Write ، Alter و Attach را پوشش می‌دهد. وقتی یک جدول ایجاد می‌شود مالک آن امتیاز کنترل آن را دریافت می‌کند. ایجاد و تغییر قوانین توسط محدودیتهای زیر اداره می‌شوند :

- ایجاد کننده یک قانون بر جدول T ، باید هر دو امتیاز Attach و Read را از جدول T داشته باشد.
- نحوهٔ عمل و شرایط قانون ایجاد شده در برابر امتیازات ایجاد کننده چک می‌شود. اگر قسمتی از شرایط یا نحوهٔ عملکرد قانون شامل عباراتی شود که ایجاد کننده اجازه اجرای آنها را نداشته، عملیات ایجاد شده مجاز شمرده نمی‌شود.
- فردی که متقاضی حذف یک قانون r بر روی جدول T است باید امتیاز کنترل و یا امتیاز Attach و کنترل را روی جدول T داشته باشد.

- فرد متقاضی تغییر قانون باید امتیاز Alter را روی قانون داشته باشد.

۳ـ۴ کنترل دسترسی احتیاطی در DBMSهای تجاری

دراین بخش، توضیح می‌دهیم که چگونه DAC در سیستم شیئی رابطه‌ای اوراکل اعمال می‌شود. در اوراکل، امتیازها به هر دو گروه کاربران و نقشها داده می‌شود. نقشها در یک سلسه مراتب سازمان دهی شده‌اند و یک نقش تمام امتیازات نقشهای زیر خود در سلسله مراتب را بدست می‌آورد. یک کاربر ممکن است مجاز برای ایفای چند نقش در یک فاصلهٔ زمانی باشد. با هر نقشی ممکن است یک کلمه عبور همراه شود. تا از استفاده غیرمجاز امتیازها جلوگیری کند.

مجموعه‌ای از پیش تعریف شده از امتیازها فراهم است که می‌تواند به هر کدام از نقشها در اوراکل تغییر داده شود. وقتی یک فرد نقشی را ایجاد می‌کند، نقش به طور خودکار با اختیارات سرپرستی[1] به ایجاد کننده داده می‌شود، که به او اجازه می دهد نقش را به نقشی دیگر بدهد یا از او بگیرد و این کار را می‌تواند با اختیارات سرپرستی یا بدون آن باشد. اوراکل همچنین گروه خاص، عمومی[2] را حمایت می‌کند که برای هر فردی قابل دسترسی است. امتیازها در پایگاه داده اوراکل به دو بخش تقسیم می‌شوند:

امتیازات سیستم و امتیازات شیئی :

امتیازات سیستم به فرد اجازه می دهد، اعمال سیستمی و یا عملی را روی داده خاصی انجام دهد. بیش از ۶۰ محدود از امتیازات سیستمی مهیاست. مثالی از این امتیازها، امتیاز حذف تاپل از هر جدولی در DB است. به علت اینکه امتیازات

---

[1]- admin options
[2]- public

سیستمی قدرتمند هستند، اغلب فقط به DBA یا تولید کنندگان برنامه‌های کاربردی داده می‌شدند. مثل نقشها این امتیازات می‌توانند با اختیارات سرپرستی داده می‌شوند و اگر شخصی امتیاز سیستمی با اختیارات سرپرستی را داشته باشد می‌تواند این امتیاز را به افراد دیگر بدهد یا از آنها بگیرد.

امتیازات شیئی به فرد اجازه می‌دهد، یک عمل خاص را روی شیئی مشخص در DB انجام دهد. امتیازات حذف یا درج تاپل در یک جدول مشخص، مثالی از این امتیازهاست. وقتی فردی شیئی را در شمای خودش ایجاد می‌کند، به طور اتوماتیک تمام امتیازات شیئی را در مورد آن شیئی مثل حق دادن این امتیاز به دیگران را دریافت می‌کند. اگر این اعطا امتیاز همراه با اختیار اعطاء آن باشد، فرد دریافت کننده می‌تواند این امتیاز را به دیگران دم بدهد. امتیاز شیئی فقط توسط شخصی که آن را اعطاء کرده بازگردانده می‌شود. بازگشت و ارجاع این امتیاز بازگشتی است.

۴ امنیت چندلایه در سیستمهای پایگاه داده‌ای

دراین بخش به شرح جنبه‌های امنیت چندلایهٔ در امنیت دائمی برای سیستمهای پایگاه داده‌ای می‌پردازیم. بخش اول به طور کلی بروی سیستمهای رابطه‌ای متمرکز است.

مسئله‌ای که باید مورد توجه قرار بگیرد این است که محصولات قابل توجه دیگری هم برای امنیت چندلایه برای سیستمهای توزیع شده[1] ، نامتجانس[2] و یکپارچه[1] است. ما دراین بخش به بحث در مورد بعضی از این پیشرفتها می‌پردازیم.

---

[1]- Distributed
[2]- hetrogeneous

۴ـ۱ مدل داده‌ای رابطه‌ای چند لایه

دریک پایگاه دادهٔ چند لایه، تمام داده‌ها به سطح امنیتی یکسانی منسوب نمی‌شوند. اگر چنین پایگاه داده‌ای براساس مدل رابطه‌ای باشد، اشیاء طبقه بندی شده ممکن است شامل کل DB، رابطه‌ها، تاپلها، صفات و عناصر داده‌ای باشد. دسترسی به چنین اشیایی توسط سیاست الزامی که در بخش ۲ صحبت آن شد، اداره می‌شود. یک DBMS چند لایه باید DB چندلایه را از دسترسی غیرمجاز یا تغییر توسط افراد در سطح امنیتی دیگر محافظت کند. یک پایگاه داده رابطه‌ای چند لایه، DB چندلایه را به عنوان مجموعه‌ای از رابطه‌ها تعریف می‌کند و چنین مدلی، مدل داده‌ای چندلایه رابطه‌ای نامیده می‌شود.

هدف طراحان پایگاه داده‌های رابطه‌ای چند لایه، تعریف نسخه‌های گوناگون از موجودیت، عمل یا حادثه یکسان در سطوح امنیتی مختلف بدون تعارض با قوانین امنیتی و جامعیتی است. یکی از مکانیزمهای ارائه شده، چند نمونه ای بودن[۲] است. این مکانیزم اجازه می‌دهد دو تاپل با کلید اولیه یکسان در یک DB در سطح مختلف امنیتی وجود داشته باشند. اگرچه، وجود دو تاپل با کلید اولیه یکسان متناقض خصوصیت جامعیت موجودیتی در مدل داده‌ای رابطه‌ای استاندارد است.

مثال زیر را در نظر بگیرید :

EMP یک رابطه با صفات زیر است :

$$DEPT \neq, NAME, SALARY, SS \neq$$

---

[1]- Fedrated
[2]- polyinstantiation

دراین رابطه، SS ≠ کلید اولیه است فرض کنیم شخصی تاپل زیر را ابتدا در EMP درج می کند :

(000, John, 60k, 120)

و سپس فرد دیگری نه از همان طبقه، تاپل زیر را وارد کند.

(000, John, 20k, 120)

اگر این تاپل پذیرفته شود تاپل چند نمونه ای[1] است.

اخیراً بحثهای بسیاری در مورد چند نمونه ای بودن وجود دارد. عده‌ای معتقدند چند نمونه‌ای بودن لازم است اگر ما قصد طراحی پایگاه داده های چندلایه با ضریب اطمینان بالا را داریم و عده‌ای براین عقیده‌اند که جامعیت DB اهمیت بیشتری دارد و چند نمونه‌ای بودن با آن در تناقض است. سیستمهایی که در اینجا بحث می‌کنیم انواع مختلفی از مدل داده‌ای چند لایه را ارائه داده اند و همه خصوصیات امنیتی را که دراین بخش در مورد آنها صحبت شد را ارضاء می‌کنند و چند نمونه‌ای بودن هم در بسیاری مدلها وجود دارد.

۴ـ۲ معماری

دراین بخش مدلهای مختلف کنترل دسترسی که برای MLS/DBMS ها تولید شده‌اند، بررسی می کنیم. در حالیکه DBMSها باید با انواع مختلف موارد امنیتی به عنوان سیستم عاملهای قابل اطمینان در تعامل باشند، خصوصیات DBMSهاست که سطح امنیتی آنها را در مقابل آنچه سیستم عاملهای سنتی انجام می‌دهند. معرفی می‌کند. به عنوان مثال اشیاء در DBMSها تمایل دارند در سایزهای

[1]- polyinstatiated

چندگانه باشند و بتوانند دانه دانه باشند. این با سیستمهایی که در آنها دانه دانه بودن وجود ندارد در تناقض است (به عنوان مثال فایلها یا سگمنت‌ها).

همچنین تفاوتهای عملیاتی آشکاری بین سیستم عاملها و DBMSها وجود دارد که چگونگی برخورد با مسئله امنیت را تحت تأثیر قرار می‌دهد. سیستم عاملها تمایل به تعامل با افرادی دارند که تلاش برای دسترسی به بعضی اشیاء دارند DBMSها اشیاء را بین کاربران تقسیم می‌کنند و برای کاربران ابزار ارتباط با اشیاء داده‌ای گوناگون را فراهم می‌کنند. همچنین به طور کلی DBMSها وابسته به سیستم عامل برای تأمین منابع هستند. بنابراین طراحی DBMSها باید در راستای چگونگی برخورد سیستم عامل با مسئله امنیت باشد.

تفاوت بین نحوهٔ عملکرد و احتیاجات امنیتی DBMSها و سیستم عاملها به این معناست که راه حلهای سنتی که برای تأمین امنیت سیستمهایی که با این سیستم عاملها بخوبی کار می‌کردند، نیاز دارند برای DBMSها تغییر داده شوند. در حال حاضر، هیچ معماری به تنهایی مورد قبول یا مورد استفاده در تولید MLS/DBMS ها نیست. گسترده وسیعی از خط مشی‌ها برای طراحی و ساخت MLS/DBMS ارائه شده است. بعضی نظریه‌ها دراین مورد عبارتند از :

- ***معماری Single-kernel***

دراین مدل کنترل دسترسی تماماً به عهدهٔ سیستم عامل است و DBMS نقشی در آن ندارد.

- ***معماری Distributed***

بر طبق این خطی مشی چندین ماشین انتهایی[1] DBMS و یک ماشین ابتدایی[2] مطمئن وجود دارد که انتهایی از طریق آن با هم در ارتباطند.

- ***معماری Trusted-Subject***

دراین خط مشی که گاهی دو هسته‌ای[3] هم نامیده می‌شود، براساس سیستم عامل عمل نمی کند و DBMS کنترل دسترسی را به عهده دارد.

- ***معماری Integrity-lock***

دراین معماری یک انتهایی DBMS مطمئن با دسترسی به تمام داده‌ها DB و یک ابتدایی نامطمئن که با کاربرها ارتباط برقرار می‌کند و یک انتهایی نامطمئن که استفاده از تکنولوژی پنهان سازی را فراهم می‌کند. دراین خط مشی مهم است که عناصر نامطمئن از هم جدا باشند. بنابراین می توان مطمئن بود که هیچ دو عنصر نامطمئن خارج از نظارت فیلتر مطمئن با هم ارتباط ندارند.

1- Back-end
2- Front-end
3- Dual-kernel

- ***معماری Extended-Kernel***

گسترشی بر مدل اول است. دراین معماری سیستم عامل همچنان دو راه حل MAC و DAC را به کار می‌برد. دراین مدل TDBMS بعضی راه‌حلهای مکمل برای فراهم کردن کنترل دسترسی را اضافه می‌کند.

۴ـ۳ محصولات تجاری

از سال ۱۹۸۸ محصولات MLS/DBMS ها تولید شده اند. گرچه تعدادی از آنها هرگز به بازار نیامدند. دراین بخش مروری داریم بر بعضی محصولات تجاری بین سالهای ۱۹۸۸ تا ۱۹۹۳.

- ***TRUDATA***

نسخهٔ اولیه TRUDATA در سالهای آخر دههٔ ۸۰، براساس معماری Integrity-lock به بازار آمد. TRUDATA از ماشین Britton lee به عنوان انتهایی نامطمئن و سیستم AT&T 3B2V/MLS به عنوان ابتدایی استفاده می‌کند.

- ***Secure Sybase***

Secure SQL Server که متعلق به شرکت Sybase است، سیستم مبتنی بر معماری Trusted-Subject با معماری CIS[1] است. در موارد اولیه قسمت مشتری معماری بروی Ultrix، SE/VMS یا SUN MLS اجراء می شد و قسمت خدمتگزار بروی Ultrix و چون Ultrix سیستم مطمئنی نبود، بنابراین محیط سیستم عاملی امنی به حساب نمی‌آمد و در نتیجه کنترل دسترسی به DBMS داده شد.

---

[1]- Client/Server

- ***Trusted Oracle***

تلاش DBMS اوراکل در بین کارهایی که انجام شده یکتاست و به گونه‌ای است که هر دو معماری Trusted-Subject,Single-kernel را ارضاء می‌کند. Single-kernel در نسخه‌های اولیه اوراکل توسط سیستم عاملها SE/VMS و HP/VX که این DB روی آنها اجراء می‌شد، تأمین می‌گردید. اوراکل مطمئن چند نمونه ای بودن برای عمل درج را حمایت می‌کند.

۵ زمینه‌های تحقیق

بسیاری محیط‌های کاربردی پیشرفته مثل کتابخانه‌های دیجیتال توزیع شده، سیستمهای اطلاعاتی ناهمگون[1] ، سیستمهای همکار[2] ، برنامه‌های کاربردی Work flow و غیره احتیاجات کنترل دسترسی بسیار زیادی دارند، به طوری که مکانیزمهای کنترل دسترسی امروزه نمی‌توانند پاسخگوی این نیازها باشد. در بسیاری موارد یا سازمان مجبور به بکارگیری سیاست خاصی برای کنترل دسترسی به صورت دستی است یا باید این سیاستها توسط برنامه کاربردی پیاده سازی شوند، که هر دو موقعیت آشکارا غیر قابل قبول است. در این بخش احتیاجات کنترل دسترسی در سه زمینه مهم امروزی مطرح می‌شود :

- ***کتابخانه های دیجیتال :***

ـ مکانیزم تشخیص انعطاف پذیر افراد.

ـ کنترل دسترسی Content-base به مولتی مدیا و داده‌های ساختار نیافته.

---

[1]- heterogeneous
[2]- Cooprative

ـ دسترسی‌های از راه دور و دسترسی به کتابخانه‌های توزیع شده.

ـ کپی کردن و استفاده کردن از اطلاعات.

- ***سیستمهای مدیریتی Work flow***

ـ کنترل دسترسی Role-base

ـ محدودیتهای اعطای مجوز بروی نقشها و کاربران.

- ***شبکه جهانی WWW***

ـ استراتژی مفید برای ذخیره سازی مجوزها.

ـ عملیات سرپرستی.

ـ مدلهای اعطای مجوز برای XML

۶ـ جمع بندی

دراین مقاله مروری داشتیم بر سیستمهای پایگاه داده ایمن. مفاهیم اساسی در کنترل دسترسی همچنین سیاستهای MAL و DAC معرفی شدند. مروری داشتیم بر سیاستهای سرپرستی ارائه شده و امنیت الزامی، مدلهای داده‌ای، معماریها، محصولات تجاری و صحبت کوتاهی شد در مورد محصولات و تمایلات امروزی.
محصولاتی دراین مقاله معرفی شدند. در مورد ساختار آنها صحبت مختصری شد، این محصولات هرساله بهبود می‌یابند. بنابراین برای اطلاعات به روزتر خواننده باید به کاتالوگ فروشندگان یا تولید کنندگان این محصولات مراجعه کند.

جهت سیستمهای پایگاه داده‌ای به سوی محصولات تحت وب پیش می رود و سیستمهای پایگاه داده دیگر سیستمهای مستقل نخواهد بود. آنها درحال یکی شدن با کاربردهای مختلفی مثل مولتی مدیا، تجارت الکترونیکی ، سیستمهای کامپیوتری سیار و کتابخانه های دیجیتال هستند. موارد امنیتی برای چنین سیستمهای بسیار مهم است. علاوه بر این امنیت در سیستمهای work folw موضوع تحقیق بسیاری از افراد است. و رقابت شدیدی برای فرموله کردن سیاستها در چنین سیستمهای وجود دارد. تکنولوژیهای جدید مثل داده کاوی [1] به حل مشکل امنیتی کمک خواهند کرد و به طور کلی DB همچنان در تکاپو خواهد بود و پیشرفت خواهد کرد و مسئله امنیت آن نمی تواند فراموش شود.

- [1] Data Mining

- مراجع

[1] Mario Piattini , and Oscar Diaz, *Advance Data base Technology and Design*, Briston: Artech House, 2000.

[2] E. Bertino, and E. Ferrari" ,Adiminstration Policies in a Ualtipolicy Authorization System ",Proc . 10th IFIP Working conf. *Data base Security*, Lake Tahoe, CA, Aug. 1997.

[3] D. Bell, and L. Lapadula, "Secure Computer systems," ESD-TR-75-306, Hanscom Air Force Base, Bedford, MA, 1975.