آموزش اعتباردهی و اعطای مجوز درSQL Server

اعتباردهی و اعطای مجوز درSQL Server

هنگام ایجاد اشیا پایگاه داده بایستی مجوزهای لازم را به صورت صریح اعطا کرده تا دردسترس کاربران قرار گیرد.
تمامی securable object ها (اشیایی که توسط یک اسم منحصر بفرد شناسایی کرد.) دارای مجوزهایی هستند که می توان با استفاده از دستورهای مبتنی بر مجوز (permission statements) آن ها را به یک principal اعطا کرد .

اصل استفاده از حساب کاربری با پایین ترین سطح دسترسی

نوشتن و توسعه ی برنامه با بهره گیری از حساب کاربری با کمترین سطح دسترسی (LUA) بخش جدایی ناپذیر یک راهبرد جامع دفاعی برای مقابله با خطرات امنیتی تلقی می شود. با پیاده سازی این رویکرد، این اطمینان حاصل می شود که کاربران از اصل (LUA) پیروی کرده و همیشه با حساب های کاربری محدود وارد شوند .(log on)کارهای مدیریتی (Administrative tasks)به وسیله ی نقش های ثابت سرور از روال عادی خود خارج می شوند، همچنین استفاده از نقش ثابت سمت سرور sysadmin بسیار محدود می باشد .
توصیه می کنیم همیشه به هنگام اعطای مجوز به کاربران پایگاه داده از اصل نام برده پیروی کنید. سعی کنید کمترین مجوزهای لازم را به یک کاربر یا نقش کاربری برای انجام رساندن کارهای معین اعطا کنید .

نکته ی امنیتی

توسعه و تست یک برنامه با بهره گیری از رویکرد LUA تا حدی فرایند توسعه و برنامه سازی را سخت تر می کند، بدین معنا که ساخت اشیا و کدنویسی در زمانی که به عنوان مدیر سامانه یا مالک پایگاه داده وارد شده اید در مقایسه با زمانی که از رویکرد LUAاستفاده می کنید، آسان تر می باشد. اما، برنامه سازی با استفاده از یک حساب کاربری با سطح دسترسی بالا (highly privileged account) ممکن است اثری که به موجب کاهش قابلیت احساس می شود را هنگامی که کاربران با کمترین سطح دسترسی سعی بر اجرای برنامه ای می کنند که برای عملکرد صحیح به مجوزهای سطح بالاتری نیاز دارد، تا حدی نسبت به قبل غیر قابل رویتر سازد.

اعطای مجوزهای غیر ضروری و بیش از حد به کاربران به منظور کسب مجدد و بازیابی قابلیت های از دسته رفته، ممکن است برنامه ی شما در برابر حملات ممکن آسیب پذیر سازد. کد نویسی و تست برنامه ی مورد نظر زمانی که توسعه دهنده با حساب کاربری LUA وارد سیستم شده، یک رویکرد منظم در راستای برنامه ریزی امنیتی پیاده سازی می کند که غافل گیرشدن و نیز وسوسه ی تخصیص مجوز های سطح بالا برای رهایی از مشکل را از میان برمی دارد. می توان برای تست برنامه از یک SQL Server login بهره جست، حتی در شرایطی که برنامه ی شما باید برای نصب یا مستقر شدن از اعتبار سنجی ویندوز استفاده کند .

مجوزهای مبتنی بر نقش

اعطای مجوز به نقش ها (بجای اعطای مجوز به کاربران) مدیریت امنیت را آسان می سازد. مجموعه مجوزهایی که به نقش های کاربری اختصاص داده می شود توسط تمامی عضوهای نقش مورد نظر به ارث برده می شود. حذف و اضافه ی کاربران از یک نقش در مقایسه با ایجاد مجدد مجموعه مجوزهای جداگانه برای تک تک کاربران آسان تر می باشد. این امکان وجود دارد که نقش ها را تودرتو کرد، با این حال استفاده ی بیش از حد از تودرتو سازی (سطوح متعدد متشکل از چندین نقش تودرتو) ممکن است منجر به کاهش سرعت و افت کارایی شود .همچنین می توان جهت آسان سازی تخصیص مجوزها، کاربران را به نقش های ثابت پایگاه داده افزود .
می توان مجوزها را در سطح schema به کاربران انتساب داد. در پی آن کاربران تمامی مجوزهای دسترسی به اشیای ایجاد شده در schema را خودکار به ارث می برند، بدین معنا که دیگر در صورت ایجاد شی جدید نیازی به اعطا مجوز نیست.

دستورات مبتنی بر مجوز

سه دستور مبتنی بر مجوز Transact-SQL در جدول زیر به صورت خلاصه شرح داده شده اند:

Permission Statement	شرح
GRANT	این دستور مجوز اعطا می کند.
REVOKE	با استفاده از این دستور می توان یک مجوز را لغو کرد. این حالت پیش فرض برای یک شی جدید می باشد. البته مجوزی که از یک کاربر یا نقش کاربری سلب شده، همواره می تواند توسط گروه ها یا نقش های دیگر که principal به آن ها اختصاص یافته، به ارث گرفته شود.
DENY	دستور DENY یک مجوز را طوری لغو می کند که امکان به ارث بری آن توسط دیگری وجود نداشته باشد. این دستور بر تمامی دیگر مجوزها اولویت و برتری دارد، با این وجود DENY به مالکان شی یا عضوهای sysadmin اعمال نمی شود.اگر شما مجوزهای (بر روی) یک شی به نقش public را DENY (کاملا لغو) کنید، مجوز شی برای تمامی کاربران به استثنای مالکان اشیا و عضوهای sysadmin، کاملا لغو می شود.

دستور GRANT قادر است به یک گروه یا نقش کاربری مجوز تخصیص دهد که این مجوز توسط کاربران پایگاه داده به ارث برده می شود. با این حال، دستور DENY بر تمامی دیگر دستورات مبتنی بر مجوز اولویت (permission statement)دارد .از این رو، کاربری که مجوز وی کلا لغو (deny)شده، دیگر نمی تواند آن را از نقش دیگر به ارث برد .

توجه :

همان طور که قبلا توضیح داده شد، نمی توان مجوز اعضای نقش ثابت سمت سرور sysadmin و مالکان شی را لغو (deny)کرد .

زنجیره های مالکیت

SQL Serverاطمینان حاصل می کند که تنها principal هایی که مجوز به آن اعطا شده ، می توانند به اشیا دسترسی داشته باشند. در مواردی که چندین شی پایگاه داده به یکدیگر دسترسی دارند، توالی مربوطه تحت عنوان یک زنجیره یا chainشناخته می شود. هنگامی که SQL Server در حال پیمایش حلقه های زنجیر است، مجوزها را به طور متفاوت در مقایسه با زمانی که به هر آیتم به صورت مجزا دسترسی پیدا می کند، مورد ارزیابی قرار می دهد .زمانی که یک شی از طریق زنجیره مورد دسترسی قرار می گیرد، SQL Serverابتدا مالک شی را با مالک شی فراخواننده (حلقه ی پیشین در زنجیره) مقایسه می کند. در صورتی که هر دو شی دارای مالک یکسان باشند، مجوزهای (بر روی) شی مورد ارجاع دیگر بررسی نمی شوند. هر زمان که یک شی، شی دیگری را مورد دسترسی قرار می دهد که مالک آن با مالک خود فرق دارد، زنجیره ی مالکیت شکسته شده و به دنبال آن SQL Server مجبور به بررسی بستر امنیتی (security context) فراخواننده می شود .

زنجیره ی مالکیت (ownership chaining) و کدهای مربوط به procedure

فرض کنید به کاربری مجوزهای اجرا بر روی یک stored procedure که کار آن انتخاب داده از یک جدول است اعطا شده. چنانچه stored procedure و جدول مورد نظر دارای مالک یکسان باشند، در این صورت دیگر لازم نیست به کاربر مجوزی بر روی جدول تخصیص داده شود، در چنین موردی حتی می توان تمامی مجوزهای کاربر را لغو (deny) کرد. اما چنانچه جدول و stored procedure دارای مالکین متفاوت باشند، SQL Serverبایستی مجوزهای کاربر مربوطه بر روی جدول را پیش از دادن اجازه ی دسترسی به اطلاعات مورد بازبینی قرار دهد .

توجه :

زنجیره ی مالکیت در مورد دستورات پویای SQL کاربرد ندارد. برای مهیا شدن زمینه ی فراخوانی پروسیجری که یک دستور SQL را اجرا می کند، به ناچار باید به فراخواننده مجوزهای برای دسترسی به جداول زیرین (underlying table)داده شود که در نهایت منجر به آسیب پذیری هر بیشتر برنامه های کاربردی شما در برابر حملات SQL Injection (SQL تزریق) می شود .SQL Server سازکارهای نوینی نظیر جعل هویت (impersonation) و امضای ماژول ها با گواهی تخصیص اعتبار (certificate)ارائه می دهد که به اعطای مجوز برای دسترسی به جداول زیرین نیاز ندارد. این ها همچنین می توانند با stored procedureهای CLR (ماشین مجازی زمان اجرای زبان مشترک) بکار روند .