

باسمه تعالی



گزارش پروژه درس سیستم‌های کامپیوتری امن

## بررسی امنیت پایگاه داده MSSQL

نگارش

سید محمد مهدی احمدپناه ۹۴۱۳۱۰۸۶

سید امیر حسین ناصرالدینی ۹۴۱۳۱۰۱۹

استاد راهنما

دکتر مهدی شجری

زمستان ۱۳۹۴

## فهرست مطالب

۵.....	مرور کلی
۸.....	مطلوب‌ها
۸.....	کاهش مساحت سطح
۱۰.....	بهترین روش‌ها برای کاهش مساحت سطح
۱۱.....	مدیریت مبتنی بر خط مشی
۱۱.....	بهترین روش‌ها برای مدیریت مبتنی بر خط مشی
۱۲.....	مدیریت و انتخاب حساب کاربری خدمت
۱۳.....	بهترین روش‌ها برای خدمت حساب‌های کاربری SQL Server
۱۳.....	بهترین روش‌های analysis utilities recommendations
۱۳.....	بهترین روش‌های برای SQL Server Patching
۱۳.....	رمزنگاری
۱۸.....	بهترین روش‌های برای رمزنگاری داده‌ها
۱۹.....	رمزنگاری SSL
۱۹.....	بهترین روش‌ها برای رمزنگاری کانال SSL
۱۹.....	کنترل دسترسی
۱۹.....	امتيازات ادمین
۲۰.....	بهترین روش‌ها برای امتيازات ادمین
۲۰.....	User-Defined Server Roles
۲۰.....	مالکیت پایگاه داده و اعتماد
۲۱.....	بهترین روش‌ها برای مالکیت پایگاه داده و اعتماد
۲۱.....	بهترین روش‌ها برای رویه‌های ذخیره‌شده در سیستم

طرح‌واره	۲۲
بهترین روش‌ها برای استفاده از طرح‌واره‌ها	۲۳
مجازشماری	۲۳
بهترین روش‌ها برای مجازشماری database object	۲۴
امنیت کاتالوگ	۲۴
بهترین روش‌ها برای امنیت کاتالوگ	۲۵
زمینه اجرا	۲۵
بهترین روش‌ها برای زمینه اجرا	۲۶
اجرای منبع داده از راه دور	۲۶
بهترین روش‌ها برای اجرای منبع داده از راه دور	۲۶
تصدیق اصالت	۲۶
حالت‌های تصدیق اصالت و لاگین‌ها	۲۶
بهترین روش‌ها برای حالت تصدیق اصالت و لاگین‌ها	۲۸
خط مشی گذرواژه	۲۹
بهترین روش‌ها برای خط مشی گذرواژه	۳۱
پایگاه‌های داده کنترل‌شده	۳۱
بهترین روش‌ها برای پایگاه‌های داده کنترل‌شده	۳۲
امنیت شبکه	۳۲
بهترین روش‌ها برای اتصال شبکه	۳۵
بازرسی	۳۶
بهترین روش‌ها برای بازرسی	۳۷
GreenSQL و مراحل پیکربندی آن	۳۸

۴۴	..... نکاتی کلی درباره کار با MSSQL
۴۵	..... خلاصه‌ای از روش‌های مناسب کلی برای امنیت SQL Server
۴۶	..... چک لیست امنیت SQL Server
۴۶	..... امنیت فیزیکی
۴۶	..... امنیت سیستم عامل
۴۶	..... نصب SQL Server
۴۷	..... حساب‌های کاربری
۴۸	..... مراجع

## مرور کلی

راهبرد دفاع در عمق به همراه لایه‌ای بودن امنیت بهترین راهکار شناخته شده برای مقابله با تهدیدات است. SQL Server از یک معماری امنیتی استفاده می‌کند که برای اجازه دسترسی دادن به مدیران و توسعه دهندگان برای ساخت یک برنامه پایگاه داده و مقابله با تهدیدات طراحی شده است. می‌توان گفت که هر نسخه از SQL Server نسبت به نسخه قبل ضمن معرفی ابزار و قابلیت‌های جدید بهبودهایی نیز داشته است. نیازمندی‌های امنیتی هر برنامه‌ای منحصر به آن برنامه است، توسعه دهندگان می‌بایست آگاه باشند از اینکه چه ترکیبی از ویژگی‌ها و توابع مناسب‌ترین ترکیب برای مقابله با تهدیدات شناخته شده و تهدیدات قابل پیش‌بینی در آینده است.

یک نمونه MSSQL شامل یک مجموعه سلسله‌مراتبی از ورودی‌هایی است که کار خود را با ارتباط با سرور آغاز می‌کنند. هر سرور مشمول چند پایگاه داده و هر پایگاه داده مشمول مجموعه‌ای از object‌هایی است که باید امنیت آنها حفظ شود. هر SQL Server یک سری مجوزهای<sup>۱</sup> مربوطه به خود را دارد. چهارچوب امنیتی SQL مدیریت دسترسی به ورودی‌ها را بر طبق تصدیق هویت<sup>۲</sup> و احراز صلاحیت<sup>۳</sup> بر عهده دارد.

تصدیق هویت: پروسه ورود به MSSQL Server به وسیله اصولی که نشان می‌دهد که شما همان شخصی که ادعا می‌کنید هستید. این پروسه یک هویت شخص را در برنامه برای او می‌سازد.

احراز صلاحیت: پروسه اینکه نشان می‌دهد که هر فرد به چه منابع دسترسی دارد و اینکه قادر به انجام چه عملیاتی بر روی آن منابع است.

این دو بخش در ادامه به صورت مفصل بررسی شده‌اند.

Microsoft SQL Server یک سیستم مدیریت پایگاه داده رابطه‌ای است که توسط شرکت مایکروسافت طراحی و توسعه داده شده است.

موتور پایگاه داده یک مجموعه سلسله‌مراتبی از موجودیت‌هایی که قرار است تحت مجوزهای امن بمانند، مدیریت می‌کند. این موجودیت‌ها به عنوان securable شناخته می‌شوند. بارزترین نوع این موجودیت‌ها

---

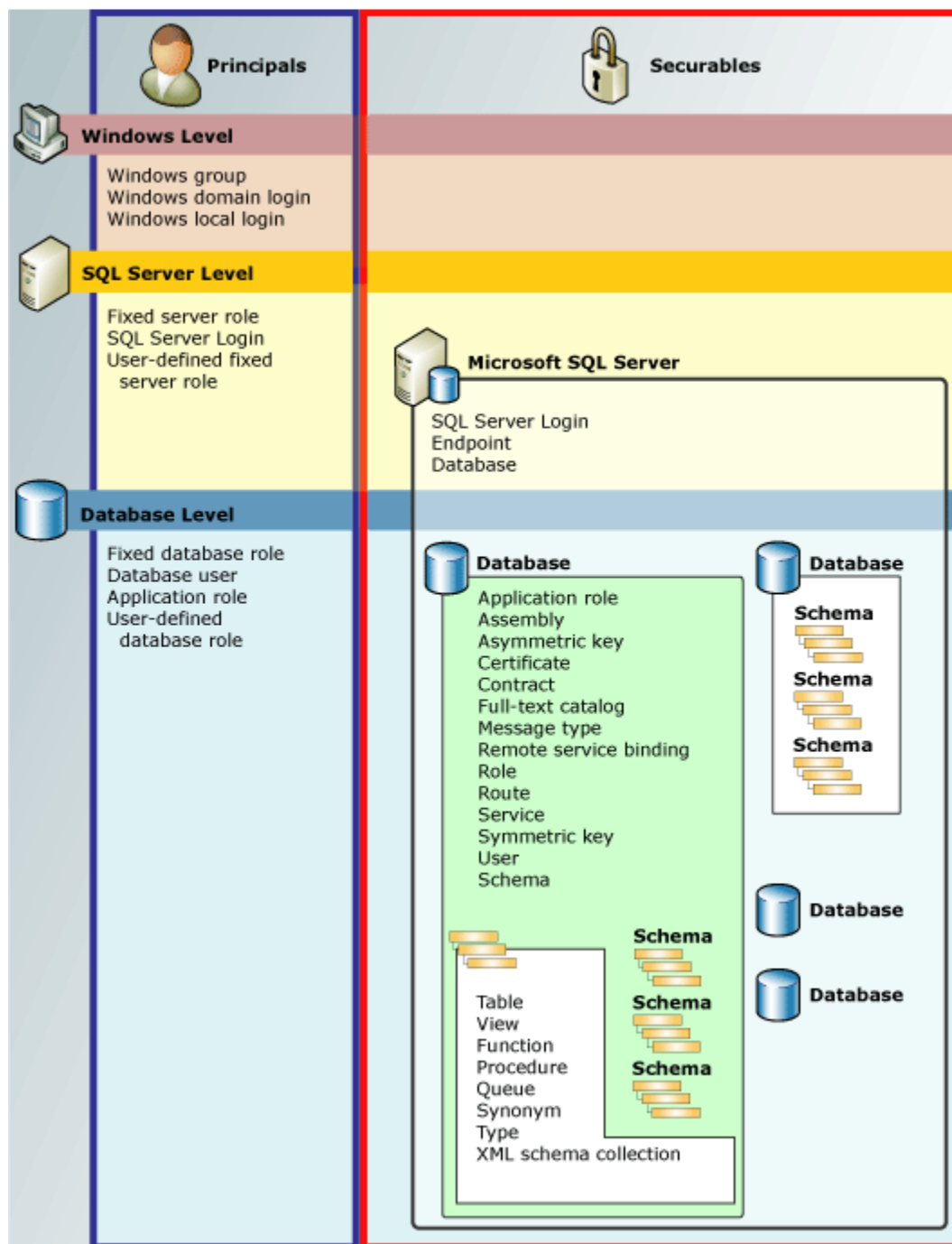
<sup>1</sup> Permission

<sup>2</sup> Authentication

<sup>3</sup> Authorization

سرورها و پایگاه داده ها هستند. MSSQL Server فعالیت های بر روی موجودیت های Securable را با راستی آزمایی مجوز های داده شده منظم می سازد.

شکل زیر روابط بین سلسله مراتب مجوز ها در موتور پایگاه داده را نمایش می دهد.



هر principle یک موجودیت است که می تواند یک درخواست برای دریافت منابع سرور MSSQL ارسال کند. همانند سایر بخش های سرور که تصدیق اصالت و احراز هویت می شوند این بخش نیز می تواند به صورت سلسله مراتبی باشد. بازه تاثیر هر principle به بازه تعرفی آنها وابستگی کامل دارد. تعریف هر principle در یکی از سه قسمت زیر قرار می گیرد:

- Windows
- Server
- Database

و هر Principle می تواند تجزیه ناپذیر (یک کاربر واحد) و یا دسته ای کاربران باشد. یک windows login میتواند نمونه ای از یک کاربر واحد باشد و یک windows group می تواند نمونه ای از یک دسته از کاربران باشد. ضمن اینکه هر کاربر یک شناسه امنیتی (SID) دارد که منحصر به خود کاربر و برای شناسایی اوست.

در MSSQL همچنین از رمزنگاری برای اختفای اطلاعات استفاده میشود. سرور برای رمزنگاری و ترجمه رمز داده ها از یک تابع نامتقارن و یا نامتقارن با مجوز استفاده می کند و همه اطلاعات را درون یک انباره داخلی مدیریت می کند. انباره از سلسله مراتب رمزنگاری برای مجوز های امنیتی استفاده می کند. می دانیم که سریعترین توابع رمزنگاری توابع متقارن هستند. این توابع برای انجام دادن اعمال مورد نظر رمز نگاری بر روی داده های عظیم مناسب هستند. کلید متقارن در این روش می تواند به وسیله یک مجوز، یک پسورد و یا سایر کلید های دیگر محافظت شود.

MSSQL از توابع متقارن زیادی برای رمزنگاری پشتیبانی می کند. این توابع شامل DES، Triple DES، RC4، 128-bit RC4، DESX، 128-bit AES، 192-bit AES و 256-bit AES هستند. و الگوریتم پیاده سازی شده از Windows Crypto API استفاده می کند.

درون یک بخش از ارتباط پایگاه داده، MSSQL Server می تواند از چندین کلید باز متقارن محافظت کند. یک کلید باز را می توان از انباره به دست آورد و برای ترجمه داده ها در دسترس قرار می گیرد. زمانی که یک بخش از داده ها ترجمه می شود نیازی نیست که مشخص شود این ترجمه توسط دام کلید انجام گرفته شده است. هر مقدار رمزنگاری شده شامل یک هویت کلیدی است که به آن GUID نیز می گویند. هویت کلید نشان دهنده کلیدی است که برای ترجمه رمز به کار می رود. موتور پایگاه داده MSSQL بایت های نشان دهنده

کلید را با یک کلید از انباره تطبیق می دهد. اگر کلید درست مورد استفاده قرار گرفته شده باشد داده ها ترجمه می شوند در غیر این صورت عمل ترجمه داده انجام نخواهد شد و مقدار NULL بازگردانده می شود.

## مطلوب‌ها

### کاهش مساحت سطح<sup>۴</sup>

نصب SQL Server 2012 حمله سطح را کمینه می کند، زیرا به طور پیش فرض ویژگی های اختیاری نصب نمی شوند. در هنگام نصب، ادمین می تواند برای نصب این موارد را انتخاب کند:

- موتور پایگاه داده
  - تکرار SQL Server
  - استخراج متن کامل و معنا برای جستجو
  - خدمات کیفیت داده
- موتور تحلیل خدمات
- خدمات گزارش دهی
- خدمات تجمیع
- خدمات داده اصلی
- کلاینت و سرور تکرار توزیع شده
- ابزارهای مختلف و کلاینت ها
- اجزای مستندسازی

یک روش بهینه آن است که ویژگی هایی از محصول که واقعا مورد نیاز است، مشخص شود و فقط آن ها نصب شوند. بعدها می توان ویژگی هایی که مورد نیاز هستند، نصب شود و فقط آن ها نصب شوند. SQL Server 2012 هیچ کد نمونه یا پایگاه داده نمونه ای را نصب نمی کند؛ البته نمونه های رسمی SQL Server بر روی سایت CodePlex وجود دارد. هر نمونه، از اصول امنیتی Microsoft Windows استفاده می کند و بر پایه اصل کمترین مجوز است.

---

<sup>4</sup> Surface Area Reduction



روشی برای امن تر کردن یک سیستم، محدود کردن تعداد ویژگی‌های اختیاری نصب و فعال شده به صورت پیش فرض است. یکی از خط مشی‌های نصب کردن SQL Server به نام «به صورت پیش فرض غیرفعال، فعال کردن در صورت لزوم» است. یکی از روش‌هایی که می‌توان مطمئن شد که خط مشی برآورده می‌شود این است که تنظیمات پیش فرض را امن کرد و استفاده از آن‌ها را ساده ساخت.

SQL Server حاوی یک ویژگی مدیریت مبتنی بر خط مشی است که به کمک آن می‌توان خط مشی‌ها را تعریف، پیاده‌سازی و اعتبارسنجی کرد و متعاقباً بهترین روش‌ها را اعمال کرد. مایکروسافت تعدادی خط مشی از پیش تعریف شده مرتبط با تنظیمات مدیریت ابزار پیکربندی مساحت سطح فراهم کرده است. این خط مشی‌ها در قالب فایل‌های XML وجود دارند و عبارتند از:

- پیکربندی مساحت سطح برای ویژگی‌های موتور پایگاه داده ۲۰۰۵ و ۲۰۰۰
- پیکربندی مساحت سطح برای ویژگی‌های موتور پایگاه داده ۲۰۰۸
- پیکربندی مساحت سطح برای نقاط انتهایی واسط خدمت
- پیکربندی مساحت سطح برای نقاط انتهایی SOAP
- پیکربندی مساحت سطح برای ویژگی‌های خدمات تحلیل
- پیکربندی مساحت سطح برای ویژگی‌های خدمات گزارش دهی ۲۰۰۵
- پیکربندی مساحت سطح برای ویژگی‌های خدمات گزارش دهی ۲۰۰۸

خط مشی‌های مدیریت مبتنی بر خط مشی پیکربندی مساحت سطح SQL Server به دو زیرمجموعه نقاط انتهایی و ویژگی‌ها تقسیم بندی می‌شوند. ویژگی‌های موتور پایگاه داده شامل خط مشی‌های از پیش تعریف شده زیر است:

- تجميع (Common Language Runtime) CLR
- استفاده از راه دور یک ارتباط تخصیص یافته به ادمین
- رویه‌های سیستم خودکار OLE
- رویه‌های سیستم برای ایمیل پایگاه داده و ایمیل SQL
- پرس و جوهای از راه دور موردی
- در دسترس بودن xp\_cmdshell
- دستیار اینترنتی SQL Server

خدماتی که بخشی از هسته موتور پایگاه داده نیستند و می‌توانند جداگانه فعال یا غیرفعال شوند، عبارتند از:

- راهنمای دایرکتوری فعال SQL Server
- عامل SQL Server
- آغازگر فیلتر Daemon کل متن
- مرورگر SQL Server
- کلاینت تکرار توزیع‌شده SQL Server
- کنترل‌کننده تکرار توزیع‌شده SQL Server
- نویسنده VSS (Volume Shadow Copy Service) SQL Server

برای داشتن پیکربندی امن‌تر، بهتر است همیشه از انتساب پورت‌های TCP/IP ایستا استفاده شود و خدمت مرورگر SQL Server غیرفعال باشد. از طرفی، اگر از VSS استفاده نمی‌شود، غیرفعال گردد.

### بهترین روش‌ها برای کاهش مساحت سطح

- فقط اجزایی نصب شوند که فوراً استفاده خواهند شد. اجزای اضافی دیگر را می‌توان در صورت نیاز، بعدها نصب کرد.
- تنها ویژگی‌های اختیاری‌ای که فوراً نیاز است، فعال شوند.
- قبل از upgrade کردن در محل، نحوه استفاده ویژگی اضافی مورد بررسی قرار گیرد و همه ویژگی‌هایی که مورد نیاز نیستند، قبل یا بعد از upgrade کردن، غیرفعال شوند.
- یک خط مشی با توجه به انتخاب‌های اتصال شبکه مجاز توسعه داده شود. برای استاندارد کردن این خط مشی از مدیریت مبتنی بر خط مشی SQL Server استفاده شود.
- یک خط مشی برای نحوه استفاده ویژگی‌های اختیاری توسعه داده شود. برای استاندارد کردن این خط مشی از مدیریت مبتنی بر خط مشی SQL Server استفاده شود. هرگونه استثنا درباره خط مشی با ذکر مصداق مستندسازی شود.
- خدمات غیر مورد نیاز توسط تنظیمات دستی خاموش یا غیرفعال شود.
- فقط اینترفیس‌های سرور شبکه که واقعاً مورد نیاز است، پیکربندی شود.

## مدیریت مبتنی بر خط مشی

مدیریت مبتنی بر خط مشی به تنهایی برای مدیریت و پیکربندی مساحت سطح استفاده نمی‌شود، اما می‌توان از آن برای تشخیص شرایط غیرمطلوب بهره برد؛ گرچه تضمین قوی‌ای برای توقف رفتار ناقض شرایط مطلوب توسط خط مشی نیست. علاوه بر خط مشی‌های پیکربندی مساحت سطح که در بالا ذکر شد، SQL Server مجموعه‌ای از خط مشی‌های بهترین روش امنیتی را شامل می‌شود. این خط مشی‌های عبارتند از:

- الگوریتم رمزنگاری کلید نامتقارن
- حقوق CmdExec امن شده
- مجوزهای مهمان
- مجوزهای Public Not Granted Server
- حالت لاگین SQL Server
- انقضای گذرواژه SQL Server
- خط مشی گذرواژه SQL Server
- رمزنگاری کلید متقارن برای پایگاه داده‌های کاربر
- رمزنگاری کلید متقارن برای پایگاه داده اصلی
- رمزنگاری کلید متقارن برای پایگاه داده‌های سیستم
- پایگاه داده قابل اعتماد

این خط مشی‌ها را می‌توان مشابه قبل پیاده‌سازی کرد یا برای کاربرد خاص شخصی‌سازی کرد تا از اعمال و امن‌سازی موارد مطلوب اطمینان حاصل کرد.

مدیریت مبتنی بر خط مشی SQL Server این اجازه را می‌دهد تا خط مشی‌ها را در قالب فایل‌های XML وارد یا صادر کرد.

## بهترین روش‌ها برای مدیریت مبتنی بر خط مشی

- یک خط مشی برای اتصال شبکه، نحوه استفاده ویژگی‌های اختیاری، و پیاده‌سازی بهترین روش‌های امنیتی SQL Server توسعه داده شود. از مدیریت مبتنی بر خط مشی SQL Server برای استاندارد کردن آن‌ها استفاده شود.

- از سرورهای مدیریت مرکزی برای استاندارد کردن و اعمال خط مشی‌های امنیتی در مقابل مجموعه‌های سرورها استفاده شود.
- از چارچوب مدیریت خط مشی Enterprise برای ادغام تاریخچه‌ها و گزارش‌گیری مجموعه خط مشی‌های در سطح enterprise استفاده شود.

## مدیریت و انتخاب حساب کاربری خدمت

SQL Server به عنوان مجموعه‌ای از خدمات ویندوز اجرا می‌شود.

زمانی که یک حساب کاربری ویندوز برای یک حساب کاربری خدمت SQL Server انتخاب می‌شود، گزینه‌های زیر وجود دارد:

- Virtual Service Account
- Managed Service Account
- Domain user that is not a Windows Admin
- Local user that is not a Windows Admin
- Network Service Account
- Local System Account
- Local user that is a Windows Admin
- Domain user that is a Windows Admin

در هنگام انتخاب گزینه‌های فوق، باید به اصل کمترین مجوز توجه داشت. ضمناً باید جداسازی حساب کاربری را در نظر گرفت. حساب‌های کاربری نباید فقط از یکدیگر متفاوت باشند، بلکه آن‌ها نباید توسط خدمت دیگری روی همان سرور مورد استفاده قرار بگیرند.

حساب‌های کاربری مجازی و حساب‌های کاربری خدمت مدیریت‌شده، دو نوع مطرح شده از حساب‌های کاربری خدمت هستند. آن‌ها برای اطمینان از جداسازی خدمات ویندوز و ایجاد provisioning، مدیریت گذرواژه و مدیریت راحت‌تر پیکربندی SPN (Service Principal Name) است.

یک حساب کاربری خدمت مدیریت‌شده، نوع خاصی از حساب کاربری دامنه است که می‌تواند به یک کامپیوتر تک منتسب شود و برای مدیریت یک خدمت مورد استفاده قرار گیرد.

حساب‌های کاربری مجازی، حساب‌های کاربری محلی مدیریت‌شده هستند که به طور خودکار فراهم و مدیریت شده‌اند.

### بهترین روش‌ها برای خدمت حساب‌های کاربری SQL Server

- در سیستم‌عامل‌های ویندوز ۷ یا R2 ۲۰۰۸، از حساب‌های کاربری خدمت مدیریت‌شده و حساب‌های کاربری مجازی سرور استفاده شود.
- در Windows Vista SP2 یا Windows 2008 SP2، از حساب کاربری Network Service یا یک حساب کاربری کاربر خاص یا حساب کاربری دامنه، به جای یک اکانت مشترک برای خدمات SQL Server استفاده شود.
- همیشه از SQL Server Configuration Manager برای تغییر حساب‌های کاربری خدمت استفاده شود.
- اگر از حساب کاربری یک کاربر یا دامنه استفاده می‌شود، در بازه‌های زمانی منظم، گذرواژه حساب کاربری تغییر داده شود.

### بهترین روش‌های analysis utilities recommendations

- اجرای SQL Server Best Practices Analyzer برای SQL Server 2008/2008 R2.
- برای فراهم کردن مدیریت امنیتی متمرکز شده از Microsoft Security Compliance Manager استفاده شود.

### بهترین روش‌های برای SQL Server Patching

- همیشه به روز باشد.
- فعال کردن به‌روزرسانی‌های خودکار، هرگاه که امکان‌پذیر بود. اما باید قبل از اعمال در سیستم، آزمون شوند.

### رمزنگاری

SQL Server رمزنگاری داده built-in دارد، هم برای یک سطح سلول و هم برای کل پایگاه داده. رمزنگاری داده نیازمند کلیدهای رمزنگاری امن و مدیریت کلید است. یک سلسله‌مراتب مدیریت کلید در SQL Server وجود دارد. هر نمونه از SQL Server یک کلید اصلی خدمت در هنگام نصب به صورت built-in تولید می‌کند؛ در واقع در اولین بار شروع پس از نصب. کلید اصلی خدمت با بهره‌گیری از کلید حساب کاربری خدمت SQL

Server و هم کلید ماشین، رمز می‌شود. هر دو عملیات رمز، از DPAPI (Data Encryption API) استفاده می‌کنند. یک ادمین پایگاه داده می‌تواند یک کلید اصلی پایگاه داده با استفاده از DDL زیر تعریف کند.

CREATE MASTER KEY

WITH ENCRYPTION BY PASSWORD = '87(HyfdlkRM?\_764#GRtj\*(NSf”\_+^\$(

این کلید به طور پیش‌فرض دوبرار رمز و ذخیره می‌شود. کلید اصلی خدمت و کلیدهای اصلی پایگاه داده می‌توانند جدا از بقیه پایگاه داده پشتیبان‌گیری و بازیابی شوند. در SQL Server 2012، از الگوریتم AES\_256 برای کلیدهای اصلی پایگاه داده و خدمت استفاده می‌شود در حالیکه در نسخه‌های قبلی، از Triple DES استفاده می‌شده است. می‌توان از DDL برای تعریف گواهی‌ها، کلیدهای نامتقارن و کلیدهای متقارن به ازای هر پایگاه داده استفاده کرد. SQL Server، از گواهی‌های X.509 استفاده می‌کند و می‌تواند آن‌ها را تولید کند.

SQL Server مکانیزم‌های زیر را برای رمزنگاری فراهم می‌کند:

Transact-SQL functions -

- کلیدهای نامتقارن

- کلیدهای متقارن

- گواهی‌نامه‌ها

Transparent Data Encryption - یک حالت خاصی از رمزنگاری با کلید متقارن است. TDE یک

پایگاه داده کامل را با کلید متقارنی که کلید رمزنگاری پایگاه داده نامیده می‌شود، رمزنگاری می‌کند. کلید رمزنگاری پایگاه داده توسط کلیدهای دیگر یا گواهی‌نامه‌ها محافظت می‌شود، که خود آن‌ها توسط کلید اصلی پایگاه داده یا یک کلید نامتقارن ذخیره‌شده در یک ماژول EKM محافظت می‌شوند.

برای استفاده از TDE باید مراحل زیر انجام شود:

- یک کلید اصلی تولید شود.

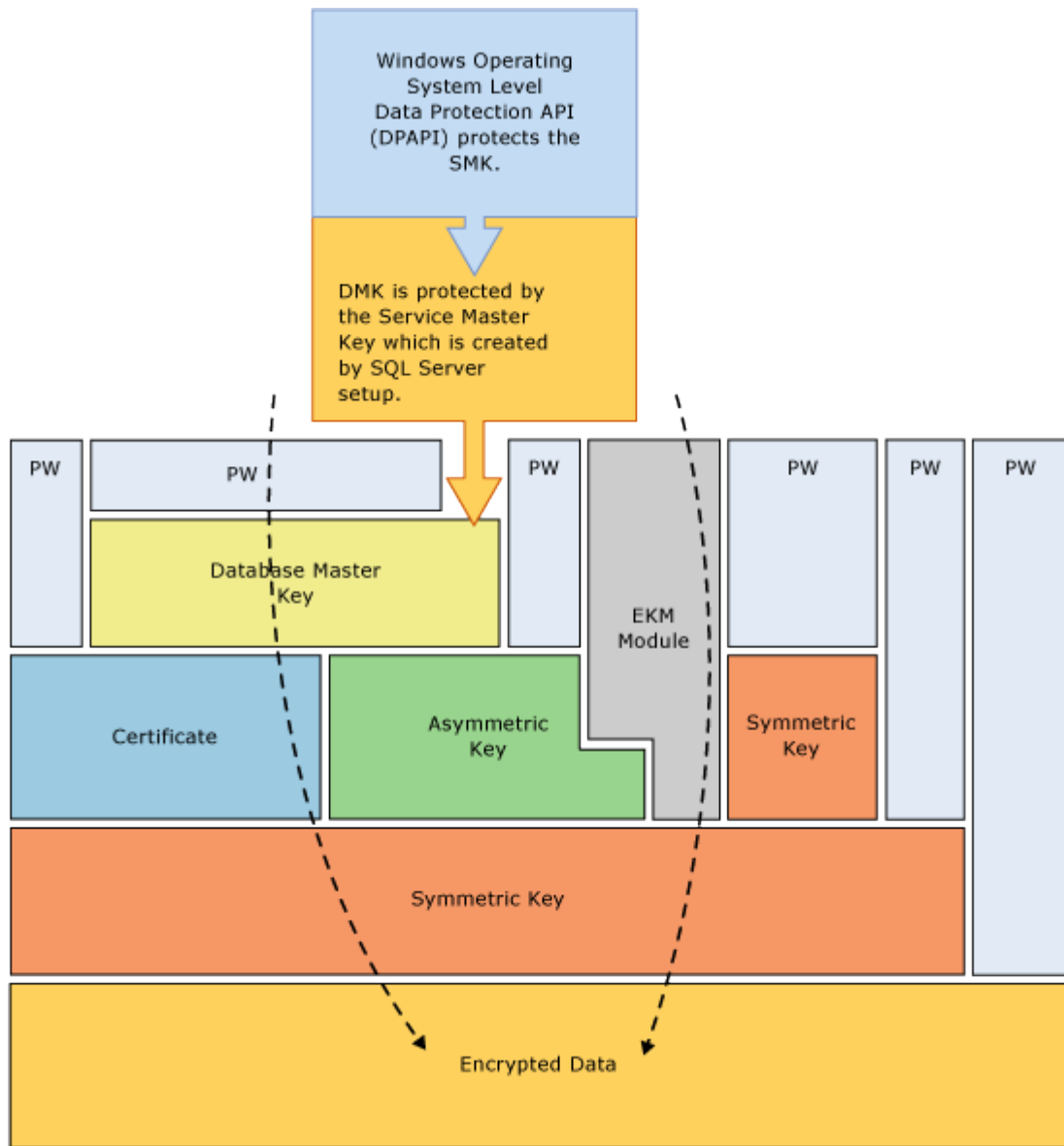
- یک گواهی‌نامه محافظت‌شده توسط کلید اصلی تولید یا دریافت شود.

- یک کلید رمزنگاری پایگاه داده ایجاد شود و آن کلید توسط گواهی‌نامه محافظت شود.

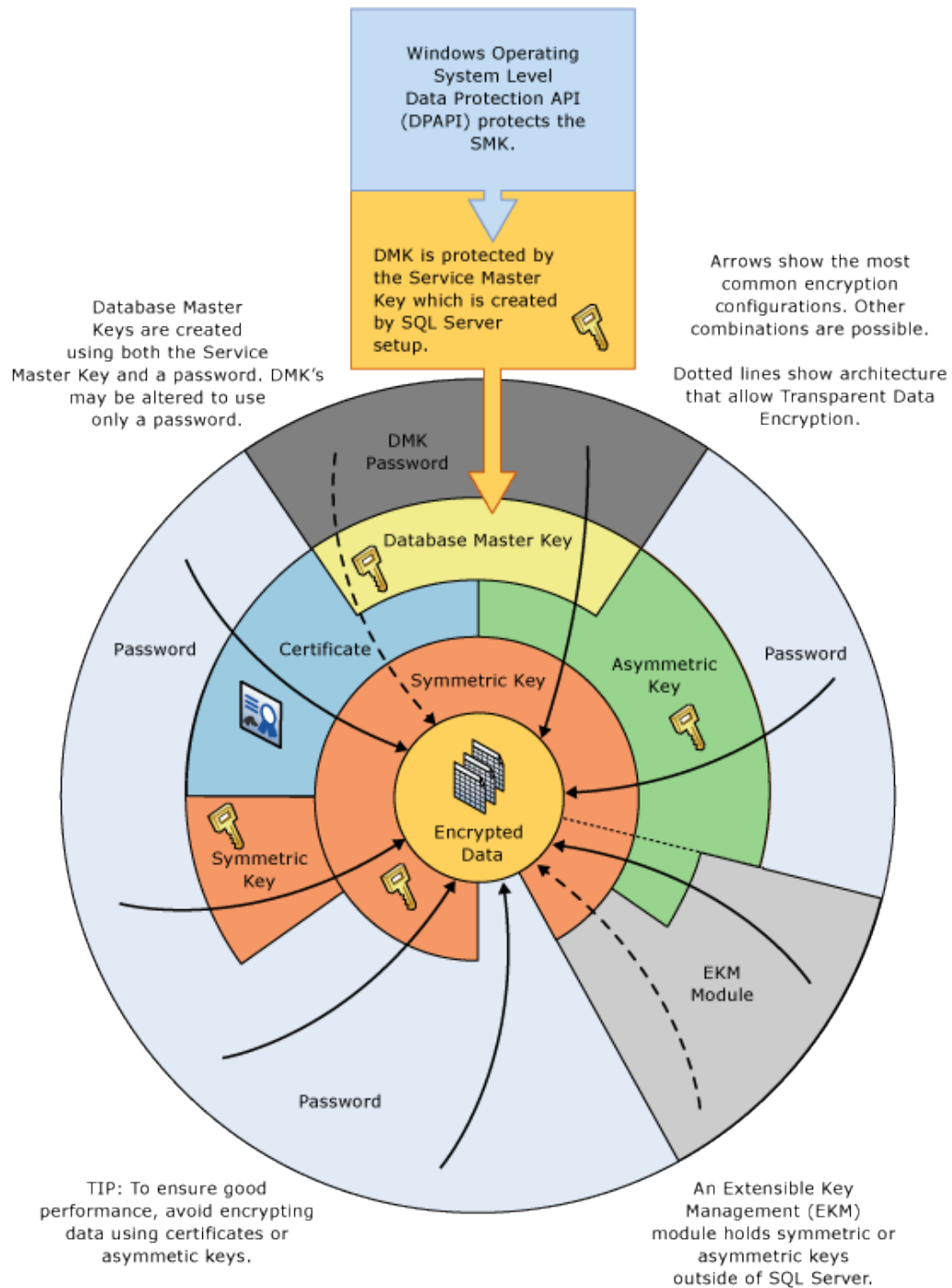
- پایگاه داده برای استفاده از رمزنگاری تنظیم شود.

انواع الگوریتم‌های رمز مختلف پشتیبانی می‌شود، مانند الگوریتم‌های DES، Triple DES و AES برای کلیدهای متقارن و RSA برای کلیدهای نامتقارن. به ازای هر الگوریتم نیز قدرت‌های مختلف پشتیبانی می‌شود. الگوریتم‌های رمز دنباله‌ای مانند RC4 و RC4\_128 تنها برای محاسبه‌پذیری وارونه، زمانی که سطح محاسبه‌پذیری پایگاه داده ۹۰ یا ۱۰۰ است، پشتیبانی می‌شود. الگوریتم‌های تعریف‌شده توسط کاربر پشتیبانی نمی‌شود. انتخاب الگوریتم کلید و طول کلید باید متناسب با حساسیت داده لحاظ شود. توابع HASHBYTES<sup>۱</sup>ی که در SQL Server 2012 پشتیبانی می‌شوند، الگوریتم‌های SHA2\_256 و SHA2\_512 است، علاوه بر الگوریتم‌های قبلی که در نسخه‌های قبلی پشتیبانی می‌شدند.

برای بعضی از قسمت‌های SQL Server می‌توان از Extensible Key Management استفاده کرد. EKM به این معناست که کلیدها توسط یک منبع خارجی مدیریت می‌شوند، مانند یک ماژول امنیتی سخت‌افزاری. منبع خارجی در SQL Server به عنوان یک cryptographic provide تلقی می‌شود. TDE از کلیدهای نامتقارنی که توسط EKM فراهم شده است، پشتیبانی می‌کند. هیچ فرم دیگری از کلید نامتقارن توسط TDE پشتیبانی نمی‌شود. به شدت توصیه می‌شود که از EKM چه برای رمزنگاری در سطح پایگاه داده و چه در سطح سلول، برای مدیریت کلید جامع و رمزنگاری مبتنی بر سخت‌افزار استفاده شود.



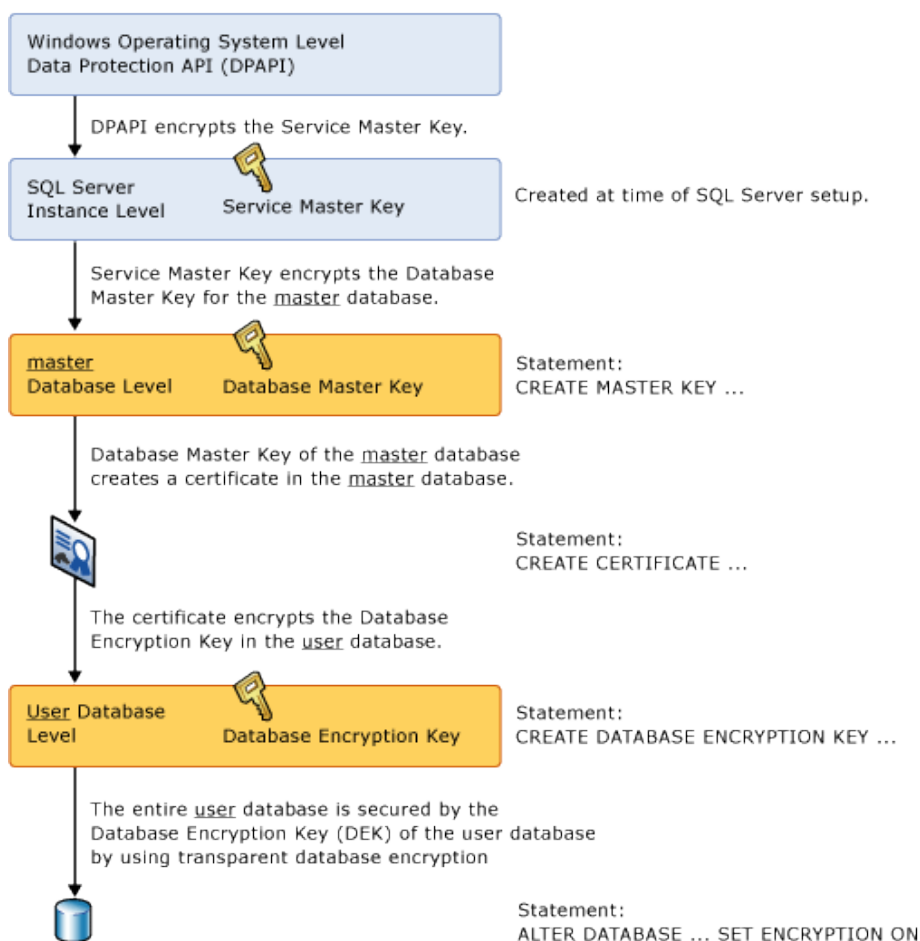




## بهترین روش‌های برای رمزنگاری داده‌ها

- داده‌های حساس و سطح بالا رمز شوند.
- از کلیدهای متقارن برای رمزکردن داده‌ها و از کلیدهای نامتقارن یا گواهی‌ها برای حفاظت از کلیدهای متقارن استفاده شود.
- برای بیشترین حد امنیت پیکربندی، از کلیدهای محافظت‌شده توسط گذرواژه استفاده شود و از رمزکردن کلید اصلی اجتناب شود.
- همواره از کلید اصلی خدمت، کلیدهای اصلی پایگاه داده و گواهی‌هایی که از گزاره‌هایی DDL مبتنی بر کلید پشتیبان‌گیری شود.
- از کلیدهای متقارن و نامتقارن پایگاه داده پشتیبان‌گیری شود.
- برای کاربردهای رمزنگاری موجود یا کاربردهای حساس به کارایی، TDE توصیه می‌شود.
- از EKM برای رمزنگاری هم سطح پایگاه داده و هم سطح سلول برای مدیریت کلید جامع و رمزنگاری مبتنی بر سخت‌افزار استفاده شود.

### Transparent Database Encryption Architecture



## رمزنگاری SSL

SQL Server از یک کانال رمزشده به دو دلیل می‌تواند استفاده کند: برای رمزکردن عبارات سری برای لاگین‌های SQL و فراهم کردن رمزنگاری انتهابه‌انتهای برای تمامی جلسات. دلیل دیگری برای استفاده از SSL، رمزکردن عبارات سری در طول فرایند لاگین برای لاگین‌های SQL است، زمانی که گذرواژه‌ای از طریق شبکه عبور کرده باشد. اگر یک گواهی‌نامه SSL در یک نمونه از SQL Server نصب شده باشد، از آن گواهی‌نامه برای رمزکردن عبارت سری استفاده می‌شود. اگر یک کواهی‌نامه SSL نصب نشده باشد، SQL Server می‌تواند یک گواهی‌نامه خودامضاشده را تولید و استفاده کند. استفاده از گواهی‌نامه خودامضاشده از حملات فردی‌درمیان منفعل جلوگیری می‌کند. استفاده از گواهی‌نامه SSL با یک مرکز صدور ریشه مورد اعتماد از حملات فردی‌درمیان فعال جلوگیری می‌کند و تصدیق اصالت دوطرفه را فراهم می‌کند.

### بهترین روش‌ها برای رمزنگاری کانال SSL

- اگر باید از لاگین‌های SQL پشتیبانی کنید، گواهی‌نامه SSL را از یک مرکز صدور مورد اعتماد، به جای گواهی‌نامه خودامضاشده، نصب کنید.
- در صورت نیاز به رمزنگاری انتهابه‌انتهای برای جلسات حساس، از allow only encrypted connections استفاده شود.
- از نسخه‌های قدیمی‌تر پروتکل TLS استفاده نشود؛ این تضمین می‌کند که اطلاعات لاگین SQL همواره رمزشده هستند.

## کنترل دسترسی

### امتیازات ادمین

SQL Server 2008 همه مجوزها را قابل اعطا می‌کند و ریزدانگی بالاتری نسبت به نسخه‌های قبل فراهم می‌کند. امتیازهای با مجوزهای عالی شامل

- اعضای نقش sysadmin سرور

- لاگین توکار sa، اگر که فعال شده باشد

- هر لاگین با مجوز control server

در هنگام نصب SQL Server، لازم است که حداقل یک لاگین ویندوز به نقش sysadmin اضافه کرد. این موضوع برای SQL Server Analysis Services و موتور پایگاه داده نیز برقرار است.

برای نگهداری حساب‌های کاربری و پاسخ‌گویی در مورد آن‌ها در پایگاه داده، از وابستگی به گروه Administrators اجتناب شود و تنها ادمین‌های مخصوصی از پایگاه داده به نقش sysadmin منتسب شوند. گزینه دیگر داشتن یک نقش مخصوص ادمین پایگاه داده در سطح سیستم عامل است. کوچک کردن تعداد ادمین‌هایی که حق دسترسی sysadmin یا Control Server دارند، باعث برطرف کردن راحت‌تر مشکلات خواهد شد.

### بهترین روش‌ها برای امتیازات ادمین

- از امتیازات ادمین فقط در زمانی که نیاز است، استفاده شود.
- تعداد ادمین‌ها را کاهش دهید.
- از ادمین‌های جداگانه متعدد، در صورت نیاز به بیش از یکی، استفاده شود.
- از وابستگی به گروه‌های ویندوز ادمین یا توکار اجتناب شود.

### User-Defined Server Roles

نقش‌های user-defined server در SQL Server 2012 معرفی شده‌اند که مشابه نقش‌های user-defined پایگاه داده عمل می‌کنند، اما با securable‌های در سطح سرور. این امکان اجازه می‌دهد تا مجموعه‌های مجوزها جمع‌آوری و به یک نقش منتسب شوند، به جای اینکه آن‌ها را تک تک به کاربرها اعطا کنیم. هر مجوز سطح سرور، مثل CREATE ANY DATABASE، ALTER ANY DATABASE، CONNECT SQL یا SHUTDOWN می‌تواند به یک نقش سرور اعطا شود. استفاده از نقش‌های سرور تعریف‌شده توسط کاربر به جای اعطای دسترسی به افراد، می‌بایست به عنوان یک بهترین روش در نظر گرفته شود.

### مالکیت پایگاه داده و اعتماد

یک نمونه SQL Server می‌تواند شامل پایگاه داده‌های کاربران متعدد باشد. هر پایگاه داده کاربر یک مالک مشخص دارد که به صورت پیش‌فرض، ایجادکننده پایگاه داده در نظر گرفته می‌شود. بنا به تعریف، اعضای نقش سرور sysadmin (شامل ادمین‌های سیستم اگر به SQL Server از طریق حساب کاربری گروه پیش‌فرض خود دسترسی دارند) مالکان پایگاه داده (DBOs) در هر پایگاه داده کاربر هستند. علاوه بر این، نقشی به عنوان

db\_owner در هر پایگاه داده کاربر وجود دارد. اعضای نقش db\_owner تقریباً امتیازات مشابه کاربر کاربر dbo را دارند.

SQL Server را می‌توان در دو حالت مجزای اجرا در نظر گرفت، که یکی از آن‌ها حالت IT department و دیگری حالت ISP است که روش‌های مختلف مدیریت SQL Server هستند. در یک IT department، نقش sysadmin سیستم مدیریت همه پایگاه‌های داده کاربر را بر عهده دارد. در محیط ISP، هر مشتری اجازه مدیریت پایگاه داده خودش را دارد و از دسترسی پایگاه داده‌های سیستم یا پایگاه داده سایر کاربران محدود شده است. به عنوان مثال، پایگاه داده‌های دو شرکت رقیب می‌تواند توسط یک ISP میزبانی شود و در یک نمونه SQL Server وجود داشته باشد.

اگر هر پایگاه داده توسط موجودیت عمومی یکسانی مالکیت و مدیریت شود، کماکان روش خوبی برای برقراری «رابطه اعتماد» با یک پایگاه داده نیست. یک رابطه اعتماد بین پایگاه داده‌ها می‌تواند با اجازه زنجیره‌سازی مالکیت پایگاه داده ضربدری یا به کمک ساختن یک پایگاه داده به عنوان مورد اعتماد از طریق خاصیت قابل اعتماد بودن برقرار شود. نمونه‌ای از برقراری خاصیت قابل اعتماد بودن اینگونه است:

```
ALTER DATABASE pubs SET TRUSTWORTHY OFF
```

### بهترین روش‌ها برای مالکیت پایگاه داده و اعتماد

- از نقش‌های user-defined server به عنوان گزینه ثانوی برای انتساب امتیازات سطح سرور به کاربران انفرادی استفاده شود.
- اگر یک ISP هستید، برای پایگاه‌داده‌ها مالکان مجزا داشته باشید؛ همه پایگاه داده‌ها نباید توسط sa مالکیت داشته باشند.
- تنظیمات Cross-Database Ownership Chaining را خاموش بگذارید، مگر آن‌که پایگاه‌داده‌های چندگانه در یک واحد مستقر شده باشند.
- از اعتماد گزینشی به جای استفاده از خاصیت قابل اعتماد بودن استفاده شود.

### بهترین روش‌ها برای رویه‌های ذخیره‌شده در سیستم

- xp\_cmdshell غیرفعال شود، مگر حتماً نیاز باشد.

- هر دو رویه ایمیل (Database Mail و SQL Mail) غیرفعال شود، مگر آن که نیاز به ارسال ایمیل از SQL Server باشد. استفاده از Database Mail بهتر است.
- از Policy-Based Management برای اعمال یک خط مشی استاندارد برای استفاده رویه گسترش یافته استفاده شود.
- هر استثنا در خط مشی استاندارد مستندسازی شود.
- رویه‌های ذخیره شده در سیستم با drop کردن پاک نشوند.
- مجوزهای پیش فرض روی objectهای سیستم تغییر داده نشود.
- همه دسترسی‌های کاربران و ادمین‌ها به رویه‌های گسترش یافته deny نشود.

## طرح‌واره

یک طرح‌واره<sup>۵</sup> یک محتوای نام‌گذاری شده برای objectهای پایگاه داده است. هر طرح‌واره یک محدوده است که متناسب با سلسله مراتب بین سطح پایگاه داده و سطح object است، و هر طرح‌واره یک مالک مشخص دارد. مالک یک طرح‌واره می‌تواند یک کاربر، یک نقش پایگاه داده، یا یک نقش برنامه کاربردی باشد.

طرح‌واره‌ها یکی از مشکلات administration که به خاطر این بود که هر شیء پایگاه داده براساس کاربری که آن را ایجاد کرده است، نام‌گذاری می‌شده است را برطرف کرده است. اگر کاربری جداولی را ایجاد می‌کرده است، نام آن جداول براساس نام کاربر بوده است. حال اگر کاربر از شرکت برود یا شغلش تغییر کند باید جداول به طور دستی، به کاربر دیگری منتقل می‌شد. اگر این انتقال صورت نمی‌گرفت، یک مشکل امنیتی پیش می‌آمد. به همین خاطر، طرح‌واره‌های مبتنی بر نقش در SQL Server 2012 به بعد مطرح شد. داشتن طرح‌واره‌های مبتنی بر نقش به این معنا نیست که هر کاربر یک مالک طرح‌واره باشد، روش خوبی است. تنها کاربرانی که نیاز به ایجاد objectهای پایگاه داده دارند، باید بتوانند این کار را انجام دهند. توانایی ساختن اشیاء به معنای مالکیت طرح‌واره نیست. اشیاء ایجاد شده در یک طرح‌واره، به طور پیش فرض تحت مالکیت مالک طرح‌واره هستند، نه ایجادکننده object.

هر کاربر یک طرح‌واره پیش فرض دارد. اگر یک شی در یک گزاره SQL توسط یک نام یک بخشی ایجاد شده یا ارجاع داده شده باشد، SQL Server ابتدا طرح‌واره پیش فرض کاربر را نگاه می‌کند. اگر آن شی در آن جا پیدا نشد،

---

<sup>5</sup> schema

SQL Server در طرح‌واره dbo جستجو می‌کند. طرح‌واره پیش‌فرض کاربر با استفاده از گزاره‌های DDL CREATE USER و ALTER USER منتسب می‌شوند.

در SQL Server 2012، کاربران نگاشت‌شده به گروه‌های ویندوزی نمی‌توانند طرح‌واره‌های پیش‌فرض داشته باشند. اگر یک کاربر ویندوز، عضوی از بیش از یک گروه ویندوزی باشد، طرح‌واره پیش‌فرض برای آن کاربر، طرح‌واره پیش‌فرض برای گروه ویندوزی‌ای است که بالاترین principal\_id را دارد.

### بهترین روش‌ها برای استفاده از طرح‌واره‌ها

- اشیای مشابه یکدیگر در یک طرح‌واره یکسان گروه‌بندی شوند.
- امنیت object پایگاه داده توسط استفاده از مالکیت و مجوزها در سطح طرح‌واره مدیریت شوند.
- برای طرح‌واره‌ها مالک‌های مجزا داشته باشیم یا از یک کاربر بدون نیاز به لاگین به عنوان مالک طرح‌واره استفاده شود.
- نیازی نیست که مالک همه طرح‌واره‌ها، dbo باشد.
- تعداد مالک‌های هر طرح‌واره کمینه شوند.
- برای کاربران نگاشت‌شده به گروه‌های ویندوزی، هر کاربر ویندوز به یک گروه ویندوزی که به پایگاه داده دسترسی دارد، محدود شوند.
- از اسامی دوبخشی برای ایجاد و دسترسی database object استفاده شود.

### مجاز‌شماری

مجاز‌شماری فرایند اعطای مجوزها روی securableها به کاربران است. در سطح سیستم عامل، securableها می‌توانند فایل‌ها، دایرکتوری‌ها، registry keyها یا چاپگرهای به‌اشتراک گذاشته‌شده باشند. در SQL Server، securableها همان database objectها هستند. طرف‌های SQL Server شامل طرف‌های سطح نمونه، مثل لاگین‌های ویندوز، لاگین‌های گروه ویندوز، لاگین‌های SQL Server و نقش‌های سرور، و طرف‌های سطح پایگاه داده، مثل کاربران، نقش‌های پایگاه داده و نقش‌های برنامه کاربردی هستند. به جز تعداد کمی از objectها که در محدوده نمونه هستند، بیشتر objectهای پایگاه داده، مثل جداول، دیدها و رویه‌ها در محدوده طرح‌واره هستند. به این معناست که مجاز‌شماری معمولاً به طرف‌های سطح پایگاه داده اعطا می‌شود.

در SQL Server، مجاز‌شماری توسط زبان دسترسی داده (DAL)، بیشتر از DDL یا DML، صورت می‌گیرد. علاوه بر دو فعل DAL، که همان GRANT و REVOKE است که توسط ISO-ANSI استاندارد شده است،

SQL Server همچنین یک فعل DENY دارد. تفاوت DENY و REVOKE زمانی است که یک کاربر، عضوی از بیش از یک طرف پایگاه داده است. اگر یک کاربر Fred عضوی از سه نقش پایگاه داده A، B و C است و نقش‌های A و B مجوز GRANTED به یک securable هستند، اگر آن مجوز از نقش C، Revoke شود، کاربر Fred کماکان به آن securable دسترسی دارد. اگر آن securable از نقش C، DENY شود، Fred نمی‌تواند به securable دسترسی داشته باشد. این باعث می‌شود تا مدیریت SQL Server شبیه مدیریت سایر بخش‌های سیستم‌های عامل خانواده ویندوز باشد.

مزیت اعطای مجوزها در سطح طرح‌واره این است که کاربر به صورت خودکار مجوزهای روی objectهای جدید ایجادشده در طرح‌واره را داراست و نیازی به اعطای صریح بعد از ساخته‌شدن شی نیست.

### بهترین روش‌ها برای مجازشماری database object

- کپسوله کردن دسترسی درون ماژول‌ها
- مدیریت مجوزها از طریق نقش‌های پایگاه داده یا گروه‌های ویندوز
- استفاده از ریزدانگی مجوز برای پیاده‌سازی اصل حداقل امتیاز
- در هیچ پایگاه داده‌ای به جز MSDB، دسترسی guest را فعال نکنید.
- از کاربران بدون لاگین‌ها به جای نقش‌های برنامه کاربردی استفاده شود.

### امنیت کاتالوگ

اطلاعات درباره پایگاه داده‌ها، جدول‌ها و سایر اشیا پایگاه داده در کاتالوگ سیستم نگهداری می‌شود. فراداده<sup>۶</sup> سیستم در جدول‌هایی در پایگاه داده master و پایگاه داده‌های کاربر وجود دارند. این جدول فراداده‌ها از طریق دیدهای فراداده ظاهر می‌شوند. در SQL Server 2000، کاتالوگ سیستم برای عموم قابل خواندن بود و نمونه پایگاه داده می‌توانست طوری پیکربندی شود که قابل نوشتن نیز باشد. در SQL Server 2005 به بعد، جداول فراداده‌های سیستم فقط خواندنی هستند و ساختار آن‌ها نیز تغییر کرده است. تنها راهی که جداول فراداده سیستم برای همه قابل خواندن باشد، این است که در حالت single-user باشد. دیدهای فراداده سیستم بازنگری شد و در قالب یک طرح‌واره خاص به نام sys قرار گرفت.

در SQL Server 2005 به بعد، همه viewهای فراداده‌ها به صورت پیش‌فرض امن شده‌اند که شامل موارد زیر است:

---

<sup>۶</sup> metadata



- دیدهای فراداده جدید (به عنوان مثال، sys.tables و sys.procedures)

- دیدهای فراداده سازگاری (مثال: sysindexes و sysobjects)

- دیدهای INFORMATION\_SCHEMA

بعضی از برنامه‌های کاربردی لیستی از اشیای پایگاه داده را از طریق رابط کاربری گرافیکی به کاربر نمایش می‌دهند. ممکن است نیاز باشد تا رابط کاربری به وسیله اجازه دادن به کاربران برای دیدن اطلاعات درباره اشیا پایگاه داده به همان شکل قبلی نگه داشته شود، در حالیکه اجازه صریح دیگری به آن شیء داده نشود. یک مجوز خاص به نام VIEW DEFINITION، برای این منظور وجود دارد. این مجوز این امکان را به فراخوان‌کننده می‌دهد تا تعریف اشیا مانند رویه‌های ذخیره‌شده را نیز ببیند.

### بهترین روش‌ها برای امنیت کاتالوگ

- دیدهای کاتالوگ به صورت پیش‌فرض امن هستند. برای امن کردن آن‌ها نیازی به کار اضافی نیست.
- اعطای VIEW DEFINITION گزینشی به شیء، طرح‌واره، پایگاه داده یا سطح سرور برای اعطای مجوز برای مشاهده فراداده سیستم بدون دادن مجوزهای اضافه دیگری.
- مرور و بازنگری برنامه‌های کاربردی قدیمی که به دسترسی به فراداده سیستم وابسته هستند، هنگامی که به نسخه‌های SQL Server بالاتر ارتقا می‌یابند.

### زمینه اجرا

SQL Server همیشه گزاره‌ها و کدهای رویه‌ای SQL را به عنوان کاربری که اخیراً لاگین شده اجرا می‌کند. به این معنا که یک رویه ذخیره‌شده به عنوان فراخوان‌کننده آن رویه ذخیره‌شده اجرا می‌شود و نه مالک آن.

انتخاب‌های موجود برای زمینه اجرا عبارتند از:

- EXECUTE AS CALLER: فراخوان‌کننده رویه (حالت پیش‌فرض)

- EXECUTE AS OWNER: مالک رویه

- EXECUTE AS SELF: ایجادکننده رویه

- EXECUTE AS 'username': یک کاربر خاص

## بهترین روش‌ها برای زمینه اجرا

- تعیین زمینه اجرا روی مازول‌ها به صورت صریح، و نه قراردادن در حالت پیش‌فرض
- استفاده از EXECUTE AS به جای SETUSER

## اجرای منبع داده از راه دور

دو راه برای اجرای کدهای رویه‌ای روی یک نمونه از راه دور SQL Server وجود دارد: پیکربندی یک تعریف سرور لینک‌شده با SQL Server از راه دور و پیکربندی یک تعریف سرور از راه دور برای آن. سرورهای از راه دور صرفاً برای حفظ سازگاری با نسخه‌های قدیمی‌تر مطرح هستند. حال آن که سرورهای لینک‌شده از نظر امنیتی دارای ریزدانی بالاتری نسبت به سرورهای از راه دور هستند.

## بهترین روش‌ها برای اجرای منبع داده از راه دور

- از رده خارج کردن هر تعریف سرور از راه دور
- جایگزینی سرورهای از راه دور با سرورهای لینک‌شده
- غیرفعال باقی گذاشتن پرس‌وجوهای موردی (مانند OPENDATASOURCE) از طریق سرورهای لینک‌شده، مگر آن‌که واقعا به آن‌ها نیاز باشد.

## تصدیق اصالت

### حالت‌های تصدیق اصالت و لاگین‌ها

SQL Server دارای دو نوع تصدیق اصالت است: تصدیق اصالت ویندوز و تصدیق اصالت حالت ترکیبی. در حالت تصدیق اصالت ویندوز، حساب‌های کاربری و گروهی مشخص شده ویندوز، می‌توانند به SQL لاگین کنند و به آن‌ها اعتماد شده است. عبارات محرمانه ویندوز در این فرایند استفاده شده است؛ یعنی عبارات محرمانه تصدیق اصالت Kerberos یا NTLM. SQL Server 2008 می‌تواند از تصدیق اصالت Kerberos با همه پروتکل‌ها استفاده کند، این در حالیست که نسخه‌های قبلی، فقط از Kerberos با پروتکل TCP/IP می‌توانستند استفاده کنند. حساب‌های کاربری ویندوز از دنباله‌ای از پیام‌های رمز شده برای تصدیق اصالت به SQL استفاده می‌کنند و هیچ گذرواژه‌ای از طریق شبکه در طول فرایند تصدیق اصالت عبور نمی‌کند. در حالت تصدیق اصالت ترکیبی، هم حساب‌های کاربری ویندوز و هم حساب‌های کاربری مشخص شده SQL Server (که به نام لاگین‌های SQL

Server شناخته می‌شوند) مجاز هستند. زمانی که لاگین‌های SQL استفاده می‌شوند، گذرواژه‌های لاگین SQL از طریق شبکه برای تصدیق اصالت عبور داده می‌شود. این باعث می‌شود که لاگین‌های SQL نسبت به لاگین‌های ویندوز از امنیت کمتری برخوردار باشند.

این یک روش مناسب است که در صورت امکان، فقط از لاگین‌های ویندوز استفاده شود. استفاده از لاگین‌های ویندوز با SQL Server، باعث به دست آمدن single sign-on و ساده‌تر شدن کارهای مدیریتی و اجرایی لاگین می‌شود. مدیریت گذرواژه از خط‌مشی‌های گذرواژه و رابط برنامه‌نویسی برنامه‌کاربردی تغییر گذرواژه ویندوز استفاده می‌کنند. کاربران، گروه‌ها و گذرواژه‌ها توسط ادمین‌های سیستم مدیریت می‌شوند و ادمین‌های پایگاه داده SQL Server فقط نگران آن هستند که کدام کاربران و گروه‌ها مجاز به دسترسی به SQL Server هستند و مدیریت مجازشماری بر عهده آنان است.

لاگین‌های SQL ممکن است باعث ایجاد محدودیت برای برنامه‌های کاربردی قدیمی شوند. زیرا در بیشتر موارد، برنامه از یک فروشنده شخص ثالث خریداری شده و تصدیق اصالت قابل تغییر نیست. کاربرد دیگر لاگین‌های SQL در برنامه‌های کاربردی کلاینت-سروری کراس‌پلت‌فرم است که در آن‌ها کلاینت‌های غیر ویندوزی، لاگین‌های ویندوزی ندارند. گرچه لاگین‌های ویندوزی دلسردکننده هستند، اما بهبودهای امنیتی‌ای در لاگین‌های SQL در SQL Server 2005 به بعد وجود دارد. این بهبودها عبارتند از قابلیت داشتن لاگین‌های SQL‌ای که از خط‌مشی گذرواژه اصولی و مبتنی بر سیستم عامل استفاده می‌کنند و دارای رمزنگاری بهتری در هنگام ارسال گذرواژه‌های SQL از طریق شبکه هستند.

در SQL Server 2005 به بعد، از گزاره‌های DDL برای ایجاد هم لاگین‌های ویندوزی و هم لاگین‌های SQL استفاده می‌شود. استفاده از گزاره CREATE LOGIN مناسب‌تر است. البته رویه‌های سیستمی ذخیره‌شده sp\_addlogin و sp\_grantlogin برای سازگاری با نسخه‌های قبلی پشتیبانی می‌شوند. SQL Server 2005 به بعد توانایی غیرفعال کردن یک لاگین یا تغییر یک نام لاگین از طریق استفاده از گزاره ALTER LOGIN را دارد.

اگر SQL Server 2005 در حالت تصدیق اصالت ویندوز به جای حالت ترکیبی نصب شود، لاگین sa غیرفعال می‌شود و بک گذرواژه تصادفی برای آن تولید می‌شود. اگر بعداً لازم بود تا تصدیق اصالت به حالت ترکیبی تغییر کند و حساب کاربری لاگین sa مجدداً فعال شود، از گذرواژه اطلاعی نخواهیم داشت. بهتر است بعد از نصب، گذرواژه sa به یک مقدار معلوم تغییر یابد.

SQL Server حاوی لاگین‌های از پیش تعریف‌شده‌ای مانند NT AUTHORITY/SYSTEM و ##MS\_PolicyEventProcessingLogin## می‌باشد. این لاگین‌ها به صورت built-in در SQL Server وجود دارند و نباید حذف نشوند.

لاگین‌ها می‌توانند علاوه بر اینکه بر اساس کاربران ویندوز است، بر اساس گروه‌های ویندوز نیز باشد. استفاده از لاگین‌های ویندوز به جای گروه ویندوز می‌تواند این امکان را ایجاد کند که تک تک کاربران ویندوز را هویت‌سنجی کرد. اگر گروه‌های ویندوز برای لاگین‌ها استفاده شوند و یک کاربر ویندوز در چند گروه ویندوزی با لاگین‌های SQL Server باشد، پایگاه داده پیش‌فرض و زبان پیش‌فرض به کمک استفاده از آن گروه ویندوزی‌ای که بالاترین principal\_id را دارد، تعیین می‌شود. بنابراین، تلاش برای نگهداری از لاگین‌های گروه ویندوزی، با توجه به داشتن پایگاه داده و زبان پیش‌فرض نسبتاً سازگار خواهد بود.

SQL Server 2005 SP2 راه‌انداز<sup>۷</sup>های لاگین را معرفی کرد. این راه‌اندازهای لاگین، کنترل دسترسی‌های بیشتری روی پایگاه داده، از طریق اجازه انجام فعالیت‌هایی در طول فرایند لاگین، فراهم می‌کند.

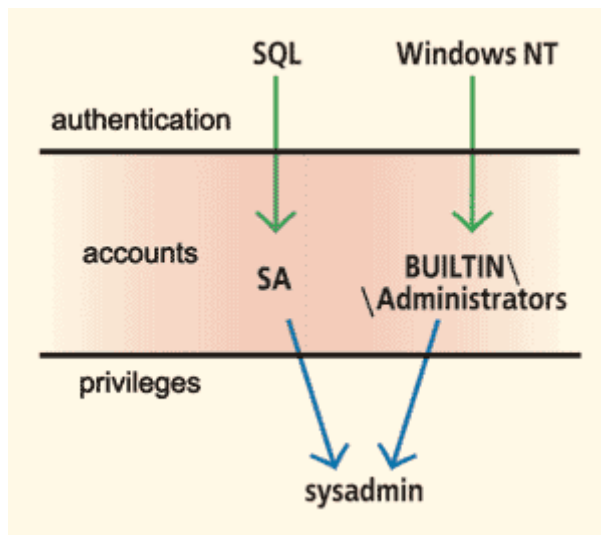
### بهترین روش‌ها برای حالت تصدیق اصالت و لاگین‌ها

- در صورت امکان، همیشه از حالت تصدیق اصالت ویندوز استفاده شود.
- از حالت تصدیق اصالت ترکیبی تنها برای برنامه‌های کاربردی قدیمی، کاربران غیر ویندوزی و کاربران از دامنه‌های untrusted استفاده شود.
- از گزاره‌های DDL لاگین استاندارد، به جای رویه‌های سیستم، استفاده شود.
- اگر قرار به استفاده از حساب کاربری sa نیست بهتر است غیرفعال شود. اگر ممکن است از حساب کاربری sa استفاده شود، گذرواژه آن حساب کاربری را به یک مقدار معلوم تغییر دهید. همواره از یک گذرواژه قوی برای حساب sa استفاده کنید و به طور مرتب، گذرواژه را تغییر دهید.
- از حساب کاربری لاگین sa برای مدیریت SQL Server استفاده نکنید. امتیازات sysadmin را به یک کاربر یا گروه معلوم انتساب دهید.
- نام حساب کاربری sa را به یک نام حساب کاربری دیگر تغییر دهید تا از حملات روی حساب کاربری sa از طریق نام جلوگیری شود.
- لاگین‌های داخلی و built-in را حذف نکنید.

---

<sup>۷</sup> trigger

- از لاگین‌های ویندوز به جای گروه ویندوز برای کنترل دسترسی به SQL Server استفاده شود.
- برای ریزدانگی بیشتر در کنترل، از راه‌اندازهای لاگین در فرایند لاگین استفاده شود.



### خط مشی گذرواژه

لاگین‌های ویندوز بر پایه خط مشی‌های لاگین مبتنی بر سیستم عامل هستند. این خط مشی‌ها با استفاده از کنترل پنل مدیریتی Domain Security Policy یا Local Security Policy قابل تنظیم هستند. خط مشی‌های لاگین به دو دسته تقسیم‌بندی می‌شوند: خط مشی‌های گذرواژه و خط مشی‌های بسته‌شدن<sup>۸</sup> حساب کاربری.

خط مشی‌های گذرواژه عبارتند از:

- اعمال تاریخچه گذرواژه
- حداقل و حداکثر سن گذرواژه
- حداقل طول گذرواژه
- گذرواژه باید نیازمندی‌های پیچیدگی را برآورده کند.
- گذرواژه‌ها باید با استفاده از رمزنگاری وارون‌پذیر ذخیره شوند. (که این تنظیمات در SQL Server به کار بسته نمی‌شود).

<sup>۸</sup> Lockout

خط مشی‌های بسته‌شدن حساب کاربری عبارتند از:

- حد آستانه بسته‌شدن حساب کاربری (تعداد لاگین‌های نامعتبر قبل از بسته‌شدن)

- مدت بسته‌شدن حساب کاربری (مقدار زمان بسته‌بودن)

- ریست‌شدن شمارنده بسته‌شدن بعد از n دقیقه

در SQL Server 2005 به بعد، لاگین‌های SQL می‌توانند از خط مشی‌های لاگین مبتنی بر سیستم عامل نیز استفاده کنند. پارامترهای CREATE LOGIN تعیین‌کننده آنست که لاگین با استفاده از خط مشی‌های سیستم عامل انجام می‌شود. این پارامترها عبارتند از:

CHECK\_POLICY -

CHECK\_EXPIRATION -

MUST\_CHANGE -

CHECK\_POLICY مشخص می‌کند که لاگین SQL باید خط مشی‌های لاگین ویندوز و خط مشی‌های بسته‌شدن حساب کاربری، به استثنای انقضای گذرواژه، را برآورده کند. به این دلیل است که اگر لاگین‌های SQL باید خط مشی انقضای گذرواژه ویندوز را برآورده کنند، برنامه‌های کاربردی وابسته باید به یک مکانیزم تغییر گذرواژه مجهز شوند. بسیاری از برنامه‌های کاربردی فعلی روشی برای تغییر گذرواژه لاگین SQL فراهم نمی‌کنند. در SQL Server 2008، هم SSMS و هم SQLCMD راهی برای تغییر گذرواژه‌های SQL Server برای لاگین‌های SQL وجود دارد. پس بهتر است در سریع‌ترین زمان ممکن، برنامه‌های کاربردی را به مکانیزم تغییر گذرواژه مجهز کرد. همچنین، داشتن تغییر گذرواژه built-in اجازه می‌دهد تا لاگین‌هایی با پارامتر MUST\_CHANGE ایجاد شوند. استفاده از این پارامتر باعث می‌شود تا کاربر در اولین بار لاگین کردن حتماً گذرواژه را تغییر دهد. ادمین‌ها باید از خط مشی‌های پیچیدگی و طول گذرواژه، و نه خط مشی‌های انقضا، که روی گذرواژه‌های استفاده‌شده با کلیدهای رمزنگاری و گذرواژه‌های مورد استفاده در لاگین‌های SQL اعمال می‌شوند، اطلاع داشته باشند.

## بهترین روش‌ها برای خط مشی گذرواژه

- یک خطی مشی گذرواژه قوی و سخت طراحی و اعمال شود که شامل خط مشی‌ای برای انقضا و پیچیدگی گذرواژه متناسب با سازمان مورد نظر باشد.
- اگر باید از لاگین‌های SQL استفاده کرد، از خط مشی‌های گذرواژه استفاده شود.
- برنامه‌های کاربردی به مکانیزم تغییر گذرواژه لاگین SQL مجهز شوند.
- پارامتر MUST\_CHANGE برای لاگین‌های جدید تنظیم شود.

## پایگاه‌های داده کنترل‌شده<sup>۹</sup>

اولین بار در SQL Server 2012 مفهوم پایگاه داده کنترل‌شده معرفی شد. یک پایگاه داده کنترل‌شده، پایگاه داده‌ای است که شامل همه تنظیمات و فراداده‌های مورد نیاز برای عملیاتش به همراه وابستگی‌هایش روی نمونه است. از منظر امنیت، یک پایگاه داده کنترل‌شده، داشتن حساب‌های کاربری یک پایگاه داده محدود شده‌اند را آسان‌تر می‌کند. SQL Server 2012 تا اندازه‌ای پایگاه‌داده‌های کنترل‌شده را پشتیبانی می‌کند. از آنجایی که پایگاه‌های داده کنترل‌شده اجازه کنترل بیشتری از طریق ادمین برنامه کاربردی دارد، پیش‌گیری‌های امنیتی بیشتری دارد.

پایگاه‌های داده از منظر امنیتی جالب هستند؛ زیرا اجازه تعریف کاربران با امتیازات تصدیق اصالت، مثل کاربرانی که می‌توانند بدون اینکه استفاده از لاگین، مستقیماً به یک پایگاه داده کنترل‌شده وارد شوند، را می‌دهند. پایگاه‌های داده کنترل‌شده از دو نوع کاربر پشتیبانی می‌کنند: کاربران ویندوز و گروه‌هایی که می‌توانند مستقیماً و بدون نیاز به لاگین‌ها به پایگاه داده متصل شوند، و کاربرانی که دارای گذرواژه‌ای هستند که گذرواژه توسط پایگاه داده، و نه نمونه آن، تصدیق اصالت می‌شود. این امکان به صورت پیش‌فرض مجاز نیست، بلکه ادمین نمونه پایگاه داده باید مشخصاً اجازه این کار را با فعال کردن گزینه پیکربندی «تصدیق اصالت پایگاه داده کنترل‌شده» بدهد. برای اجازه ندادن به یک پایگاه داده کنترل‌شده خاص و اجازه دادن به بقیه پایگاه‌های کنترل‌شده، یک راه‌انداز لاگین می‌تواند مورد استفاده قرار بگیرد.

---

<sup>۹</sup> contained database

اجازه دادن به کاربران برقراری ارتباط مستقیم با پایگاه داده، سطح تهدید موثر بعضی مجوزهای موجود را تغییر می‌دهد. برای مثال، مجوز ALRER ANY USER در یک پایگاه داده کنترل‌شده، مجوز اضافه کردن دسترسی مبتنی بر کاربر را به نمونه می‌دهد.

به دلیل نگهداری درهم‌سازی شده گذرواژه‌ها در پایگاه داده، و نه master، ذخیره می‌شود، هر کسی که به فایل پایگاه داده دسترسی یابد، می‌تواند حمله لغت‌نامه را به یک نمونه پایگاه داده بدون بازرسی انجام دهد.

ممکن است در حالی که یک لاگین و کاربر پایگاه داده کنترل‌شده دارای نام یکسانی باشند، ناسازگاری پیش آید. برای برطرف کردن این ناسازگاری، قاعده آنست که اگر یک کاتالوگ اولیه در رشته ارتباطی و پایگاه داده کنترل‌شده‌اش مشخص شده باشد، دسترسی با بررسی کاربر مبتنی بر principal، و نه لاگین، داده می‌شود. علاوه بر این، اعضای دارای نقش sysadmin نباید از کاتالوگ اولیه در یک رشته ارتباطی استفاده کنند.

### بهترین روش‌ها برای پایگاه‌های داده کنترل‌شده

- از تنظیمات پیش‌فرض (خاموش) برای تصدیق اصالت پایگاه داده کنترل‌شده استفاده شود و تنها زمانی که نیاز است، فعال شود.
- از نسخه‌های پشتیبان پایگاه‌های داده کنترل‌شده با استفاده از گذرواژه‌ها حفاظت شود.
- توانایی‌های کاربران و ماژول‌های موجود در پایگاه داده کنترل‌شده را بازرسی کنید.
- اگر تصدیق اصالت پایگاه داده کنترل‌شده مجاز باشد، لاگین‌هایی که توانایی محدودکردن را دارند، بازرسی شوند.
- بر روی پایگاه‌های داده که یک نمونه به‌اشتراک‌گذاری شده با پایگاه‌های داده کنترل‌شده دارد، حساب کاربری مهمان غیرفعال شود.
- از ناسازگاری نام‌گذاری login/user-with-login اجتناب شود.
- اگر تصدیق اصالت پایگاه داده کنترل‌شده مجاز است، رشته‌های ارتباطی با کاتالوگ اولیه وجود نداشته باشد.

### امنیت شبکه

برای اتصال به پایگاه داده SQL Server به یک پروتکل شبکه استاندارد نیاز است. هیچ ارتباط داخلی برای دور زدن شبکه وجود ندارد. به عنوان قسمتی از مراحل نصب SQL Server 2012، اگر فایروال ویندوز روی



ماشین سرور فعال نباشد، یک پیغام هشدار نمایش داده می‌شود. این یک روش مناسب برای امنیت شبکه است که فایروال ویندوز فعال باشد و پروتکل‌ها و درگاه‌های شبکه به حداقل مورد نیاز برای عملیات SQL Server محدود شود.

SQL Server 2005 یک تجزیدی برای مدیریت هر کانال ارتباطی معرفی کرده است و آن این است که نقاط ورودی به نمونه SQL Server، همگی به عنوان نقاط انتهایی بیان می‌شوند. برای پروتکل‌های ارتباطی کلاینت شبکه زیر، نقاط انتهایی وجود دارد.

Shared Memory –

Named Pipes –

TCP/IP –

Dedicated Administrator Connection

علاوه بر این، نقاط انتهایی ممکن است برای کاربردهای زیر تعریف شوند تا اجازه دسترسی به نمونه SQL Server داده شود:

Service Broker –

Database Mirroring –

HTTP Web Services – (که این نقاط انتهایی در SQL Server 2012 حذف شده‌اند.)

مثال زیر برای ایجاد یک نقطه انتهایی برای Service Broker است:

```
CREATE ENDPOINT BrokerEndpoint_SQLDEV01
```

```
AS TCP
```

```
( LISTENER_PORT = 4022 )
```

```
FOR SERVICE_BROKER
```

```
( AUTHENTICATION = WINDOWS )
```

مطابق با خط مشی کلی «خاموش به طور پیش فرض، فقط در موقع نیاز فعال»، هیچ نقطه انتهایی Service HTTP Broker یا Database Mirroring ای هنگام نصب SQL Server 2012 ایجاد نمی شود و همچنین در نسخه ای از SQL Server 2012، به طور پیش فرض پروتکل های Named Pipes و TCP/IP غیرفعال هستند و تنها Shared Memory به صورت پیش فرض فعال است.

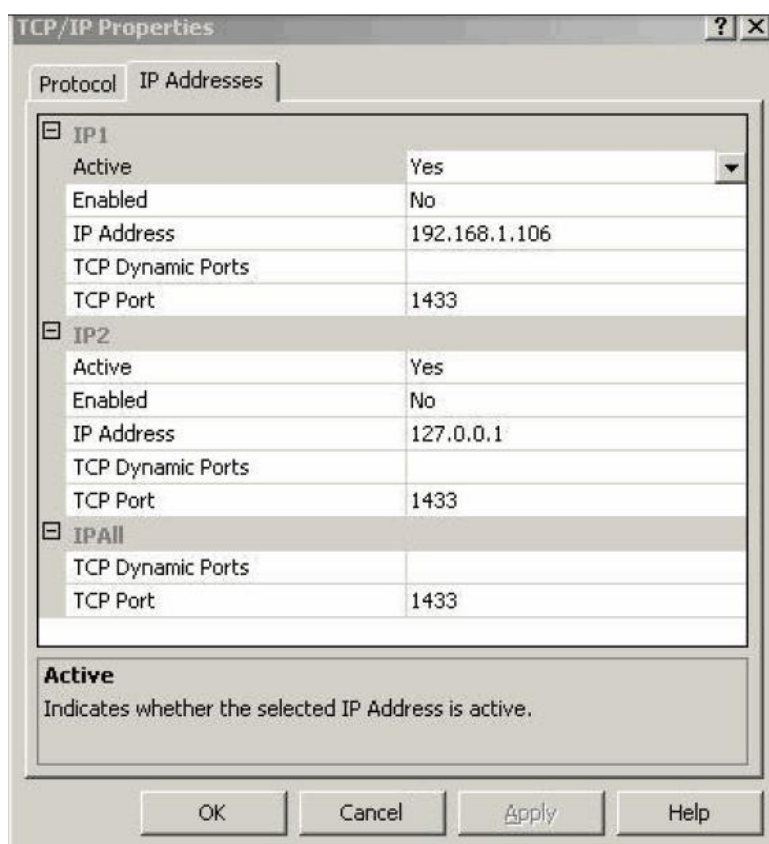
روش مناسب این است که تنها پروتکل هایی که مورد نیاز است، فعال شوند. به طور مثال، اگر TCP/IP کافی است، نیازی به فعال سازی پروتکل Named Pipes نیست.

درگاه های اضافی برای فعال سازی پروتکل NETBIOS (UDP/137, UDP/138, TCP/139) و پروتکل SMB یا Server Message Block (TCP/139, TCP/445) مورد نیاز است. این درگاه ها باید به طور پیش فرض غیرفعال باشند و تنها در زمانی که نیاز است، فعال شوند. به عنوان مثال، پروتکل SMB زمانی نیاز است که از یک SMB File Share به عنوان گزینه ذخیره سازی استفاده شود.

اگرچه مدیریت نقاط انتهایی توسط DDL نیز قابل انجام است، این فرایند مدیریت با استفاده از ابزار SQL Server Configuration Manager و Policy-Based Management راحت تر شده است. Configuration Manager پیکربندی با ریزدانی برای پروتکل های سرور فراهم می کند. با SQL Server Configuration Manager، می توان کارهای زیر را انجام داد:

- انتخاب یک گواهی نامه برای رمزنگاری SSL
- تنها ارتباط های رمزنگاری از کلاینت ها مجاز باشد.
- مخفی سازی یک نمونه SQL Server از رابط های برنامه نویسی کاربردی مربوط به شمارش سرور.
- فعال و غیرفعال کردن پروتکل های TCP/IP، Shared Memory و Named Pipes
- پیکربندی نام pipe برای هر نمونه SQL Server
- پیکربندی یک شماره درگاه TCP/IP برای نمونه هایی که روی ارتباط های TCP/IP گوش می دهند.
- انتخاب استفاده یا عدم استفاده از انتساب درگاه پویای TCP/IP برای نمونه های نام گذاری شده.
- پیکربندی حفاظت گسترش یافته برای service binding و channel binding در SQL Server 2012.

به عنوان مثال، نمایی از صفحه پیکربندی آدرس‌های TCP/IP در SQL Server Configuration Manager در شکل زیر آمده است.



در SQL Server 2008، می‌توان از مجوزهای GRANT، REVOKE یا DENY برای اتصال به یک نقطه انتهایی مشخص روی هر پایه به ازای لاگین استفاده کرد. به صورت پیش‌فرض، همه لاگین‌ها دارای مجوز GRANT بر روی نقاط انتهایی Shared Memory، Named Pipes و TCP/IP هستند. باید مشخصا به کاربران مجوز CONNECT به دیگر نقاط انتهایی اعطا شود و هیچ کاربری به طور پیش‌فرض، این امتیاز را ندارد. نمونه‌ای از اعطای این مجوز چنین است:

```
GRANT CONNECT ON MyHTTPEndPoint TO MyDomain\Accounting
```

### بهترین روش‌ها برای اتصال شبکه

- فعال‌سازی فایروال ویندوز و محدودسازی پروتکل‌های شبکه پشتیبانی‌شده.
- پروتکل‌های شبکه فعال نشوند، مگر در صورت لزوم.
- غیرفعال بودن پروتکل‌های NETBIOS و SMB، مگر در مواقع نیاز.

- یک سرور در حال اجرای SQL Server، در شبکه اینترنت عمومی در معرض دید قرار داده نشود.
- نمونه‌های SQL Server دارای نام برای استفاده از انتساب درگاه‌های مشخص برای TCP/IP پیکربندی شود، به جای درگاه‌های پویا.
- اگر کلاینت و سیستم عامل پشتیبانی می‌کند، از حفاظت گسترش‌یافته در SQL Server 2012 استفاده شود.
- مجوز CONNECT را فقط به نقاط انتهایی به لاگینی اعطا شود که به آن‌ها نیاز است. صریحاً مجوز CONNECT به نقاط انتهایی که کاربران یا گروه‌های آن‌ها نیاز ندارند، رد شود.

## بازرسی

SQL Server 2012 بازرسی داخلی‌ای که در موتور پایگاه‌داده در SQL Server 2008 اضافه شده بود، را گسترش داده است. ویژگی SQL Server Audit همه توانایی‌های راه‌حل‌های بازرسی SQL Server 2005 را دارد و بهبودهایی نظیر انعطاف‌پذیری در هدف‌های داده مورد بازرسی و بازرسی با ریزدانگی بالا را فراهم کرده است. نسخه‌های اولیه SQL Server از بازرسی لاگین، بازرسی مبتنی بر راه‌انداز، و بازرسی رویداد پشتیبانی می‌کند.

ویژگی SQL Server 2008 Audit از بازرسی مبتنی بر trace به عنوان راهکار بازرسی استفاده می‌کند. این ویژگی برای رسیدن به اهداف زیر طراحی شده است:

- امنیت: ویژگی بازرسی، و object‌های آن باید به درستی امن باشند.

- کارایی: تأثیر کارایی باید کمینه باشد.

- مدیریت: ویژگی بازرسی باید برای مدیریت آسان باشد.

- قابلیت شناسایی: سوال‌های اساسی درباره بازرسی باید به راحتی قابل پاسخ‌گویی باشد.

بازرسی SQL Server 2012 می‌تواند از یک فایل به عنوان هدف بازرسی استفاده کند و همچنین می‌تواند از Windows Application Log یا Windows Security Log بهره بگیرد. Windows Security Log در برابر دستکاری و عدم انکار مقاوم است، گرچه استفاده از آن توسط یک group policy object کنترل شده است. بازرسی با استفاده از Windows Security می‌تواند با Audit Collection Service (ACS) در Microsoft

System Center Operations Manager، که می‌تواند اطلاعات بازرسی را از ماشین‌های متعددی به طور امن جمع‌آوری و گزارش یکپارچه‌ای تولید کند، تجمیع شود.

فراداده SQL Server 2008 Audit با استفاده از DDL قابل تعریف است و بنابراین به کمک مجوزهای SQL Server استاندارد قابل مدیریت است. تغییرات در فراداده بازرسی، مانند فعال و غیرفعال کردن جلسه‌های بازرسی، نیز بازرسی می‌شوند.

اطلاعات بازرسی به صورت باینری در فایل‌های هدف نوشته می‌شوند و با تابع جدول‌ارزشی `fn_get_audit_file()` قابل خواندن است. اطلاعات بازرسی نوشته‌شده در Windows Log، با هر ابزاری که می‌تواند Windows Log را بخواند، مانند Windows Event Viewer، قابل خواندن است. هم فایل و هم اطلاعات بازرسی مبتنی بر Windows Log مستقیماً توسط SQL Server Management Studio قابل خواندن است.

### بهترین روش‌ها برای بازرسی

- بازرسی یک عمل scenario-specific است. نیاز به بازرسی و سربار تولید داده اضافی باید متوازن باشد.
- برای امنیت و ریزدانگی بیشتر از ویژگی SQL Server 2012 Audit استفاده شود.
- اگر داده‌های بسیار حساس ذخیره می‌شوند، لاگین‌های موفق نیز علاوه بر لاگین‌های ناموفق باید ثبت شوند.
- DDL و رویدادهای سرور مشخص توسط دنباله‌ای از رویدادها یا اطلاع‌های رویداد ثبت شوند.
- DML می‌تواند توسط دنباله رویدادها یا SQL Server Audit بازرسی شود.

## GreenSQL و مراحل پیکربندی آن

GreenSQL یک فایروال پایگاه داده متن باز است که برای محافظت از پایگاه داده‌ها در برابر حملات SQL Injection استفاده می‌شود. GreenSQL به عنوان یک پروکسی برای دستورات SQL عمل می‌کند و از پایگاه داده MySQL پشتیبانی می‌کند. منطق این برنامه بر اساس ارزیابی دستورات SQL با استفاده از یک ماتریس امتیازدهی ریسک و همچنین بلاک کردن دستورات مدیریتی و اجرایی پایگاه داده مثل CREATE و DROP است.



فایروال پایگاه داده GreenSQL در چند حالت مختلف می‌تواند مورد استفاده قرار گیرد:

- حالت شبیه‌سازی (Database IDS)

- بلاک کردن دستورات مشکوک (Database IPS)

- حالت یادگیرنده

- محافظت فعال در برابر کوئری‌های ناشناخته (Database Firewall)

GreenSQL کوئری‌های غیرمجاز را با استفاده از پیدا کردن دستورات مدیریتی و حساس SQL و محاسبه ریسک کوئری می‌یابد. GreenSQL از موتور تطبیق الگو برای پیدا کردن دستوراتی که غیرمجاز در نظر گرفته شده‌اند، استفاده می‌کند. یک زیرسیستم مبتنی بر signature است. دستورات مدیریتی پایگاه داده، دستوراتی که سعی در تغییر ساختار پایگاه داده دارند و دستوراتی که برای دسترسی به فایل‌های سیستم استفاده می‌شوند، از جمله

دستورات غیرمجاز تلقی می‌شوند. همچنین مدیر سیستم می‌تواند کوئری مورد نظرش را با اضافه کردن آن به لیست سفید، مجاز کرده و یا فایل پیکربندی را با لیستی از الگوهای غیرمجاز اصلاح کند.

GreenSQL برای هر کوئری، ریسک آن را محاسبه می‌کند که یک زیرسیستم تشخیص ناهنجاری است. بعد از محاسبه ریسک کوئری، GreenSQL می‌تواند کوئری را بلاک کند و یا بسته به حالت برنامه، فقط یک پیغام خطر ایجاد کند. از چندین عامل برای محاسبه ریسک استفاده می‌شود. مانند:

- دسترسی به جداول حساس، ریسک را افزایش می‌دهد.

- وجود کامنت‌ها در دستورات SQL، ریسک کوئری را بالا می‌برد.

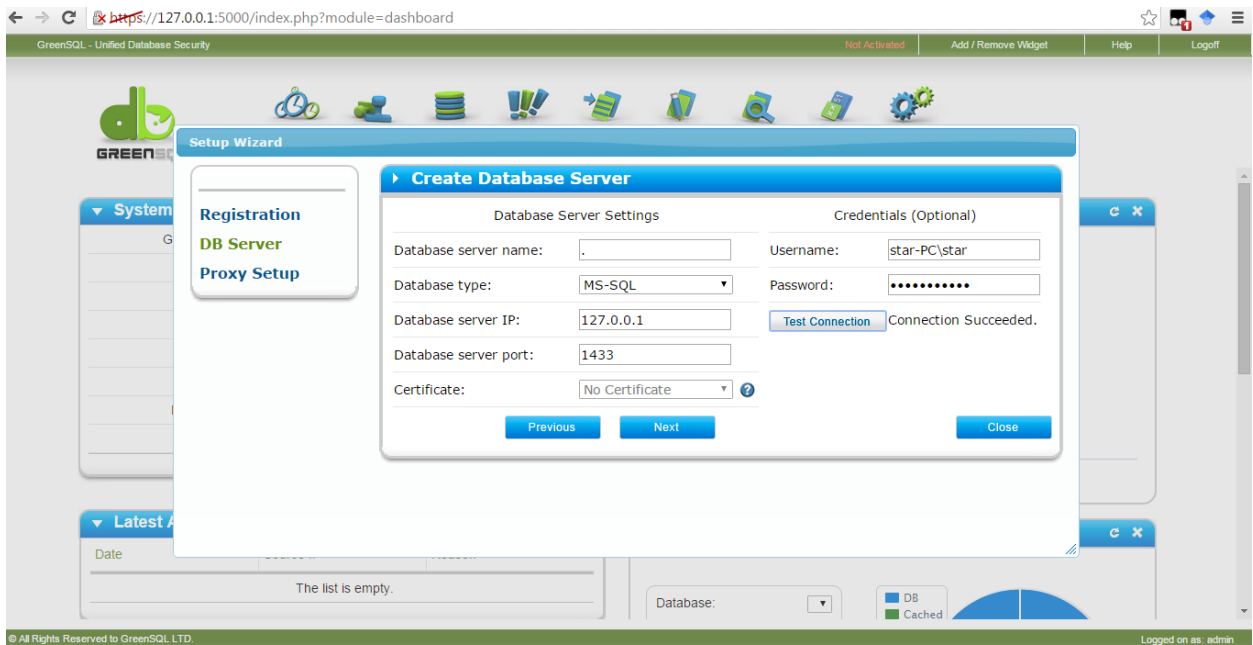
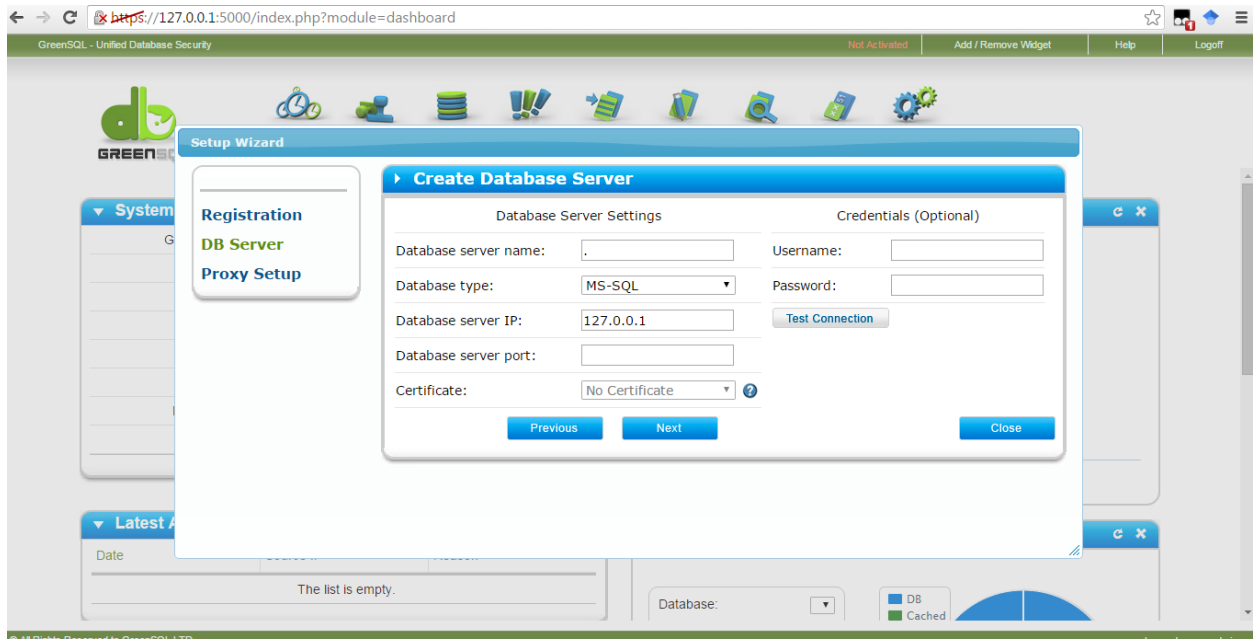
- استفاده از یک رشته گذرواژه خالی

- وجود نشانه or در کنار کوئری‌ها

- وجود یک عبارت SQL که همیشه مقدار صحیحی را برگرداند.

- مقایسه مقادیر ثابت

در ادامه، تصاویری از نحوه پیکربندی GreenSQL آمده است.





← → ↻ <https://127.0.0.1:5000/index.php?module=policy/list/edit&ruleid=1> ☆

GreenSQL - Unified Database Security Not Activated [Help](#) [Logout](#)

[Dashboard](#) [Policies](#) [Databases](#) [Alerts](#) [Logs](#) [Reports](#) [Auditing](#) [Masking](#) [System](#)

[Hide](#)

**Policies**

---

[Policy](#)

[Objects](#)

[Risk Profiles](#)

[Query Groups](#)

### Edit Rule #1

Rule Type:  Database:  [New](#)

Firewall Type:  Proxy:

Source IP:  [New](#) Table:  [New](#)

Database User:  [New](#) Action:

Application Name:  Blocking action:

Schedule:  [New](#) Logging:

Comment:

© All Rights Reserved to GreenSQL LTD. Logged on as: admin

← → ↻ <https://127.0.0.1:5000/index.php?module=policy/list/edit&ruleid=2> ☆

GreenSQL - Unified Database Security Not Activated [Help](#) [Logout](#)

[Dashboard](#) [Policies](#) [Databases](#) [Alerts](#) [Logs](#) [Reports](#) [Auditing](#) [Masking](#) [System](#)

[Hide](#)

**Policies**

---

[Policy](#)

[Objects](#)

[Risk Profiles](#)

[Query Groups](#)

### Edit Rule #2

Rule Type:  Database:  [New](#)

Firewall Type:  Proxy:

Source IP:  [New](#) Query Groups:

Database User:  [New](#) Action:

Application Name:  Blocking action:

Schedule:  [New](#) Logging:

Comment:

© All Rights Reserved to GreenSQL LTD. Logged on as: admin

← → ↻ <https://127.0.0.1:5000/index.php?module=databases/proxies/dbservers/edit&dbserverid=3> ☆

GreenSQL - Unified Database Security Not Activated [Help](#) [Logoff](#)

[Dashboard](#) [Policies](#) [Databases](#) [Alerts](#) [Logs](#) [Reports](#) [Auditing](#) [Masking](#) [System](#)

[Hide](#)

**Databases**

[Databases](#)

[Proxies](#)

[Proxies](#)

[Database Servers](#)

[Certificates](#)

### Edit Database Server

Database Server Settings	Credentials (Optional)
Database server name: <input type="text" value="Server1"/>	Username: <input type="text" value="star-pc\star"/>
Database type: <input type="text" value="MS-SQL"/>	Password: <input type="password" value="*****"/>
Database server IP: <input type="text" value="127.0.0.1"/>	<a href="#">Test Connection</a> Connection Succeeded.
Database server port: <input type="text" value="1433"/>	
Certificate: <input type="text" value="No Certificate"/>	

[Update](#) [Cancel](#)

© All Rights Reserved to GreenSQL LTD. Logged on as: admin

← → ↻ <https://127.0.0.1:5000/index.php?module=databases/proxies/list/edit&proxyid=3> ☆

GreenSQL - Unified Database Security Not Activated [Help](#) [Logoff](#)

[Dashboard](#) [Policies](#) [Databases](#) [Alerts](#) [Logs](#) [Reports](#) [Auditing](#) [Masking](#) [System](#)

[Hide](#)

**Databases**

[Databases](#)

[Proxies](#)

[Proxies](#)

[Database Servers](#)

[Certificates](#)

### Edit Proxy

Proxy Settings	Database Server
Front-end name: <input type="text" value="Proxy1"/>	Primary server: <input type="text" value="Server1"/>
Front-end IP: <input type="text" value="0.0.0.0"/>	<input checked="" type="checkbox"/> Enable fallback database <a href="#">Switch</a>
Front-end port: <input type="text" value="1433"/>	Secondary server: <input type="text" value="BackupServer"/>
	Number of retries: <input type="text" value="20"/>

[Update](#) [Cancel](#)

© All Rights Reserved to GreenSQL LTD. Logged on as: admin

← → ↻

https://127.0.0.1:5000/index.php?module=databases/proxies/list

☆

⚙


☰










GreenSQL - Unified Database Security

Not Activated

Help


Logout





Dashboard Policies Databases Alerts Logs Reports Auditing Masking System

Hide



Databases

Databases

Proxies

Proxies



Database Servers

Certificates

Create New

Customize

Proxies

Active	ID	Proxy Name	Database Server	Database Type	Status	
<input checked="" type="checkbox"/>	3	Proxy1	Server1	MS-SQL	Inactive	 

1

Per Page: 10

All Rights Reserved to GreenSQL LTD

Logout/admin

## نکاتی کلی درباره کار با MSSQL

- بهتر است Database Zone و Application Zone در شبکه از یکدیگر جدا باشد و سروری که در آن پایگاه داده وجود دارد، در Database Zone باشد. معمولاً این لایه، امن‌ترین لایه شبکه است و تنها درگاه SQL بین این دو ناحیه روی فایروال باز باشد.

- سطح دسترسی کاربران از اهمیت بالایی برخوردار است. برای اتصال نرم‌افزار به پایگاه داده باید کاربر مجزایی تعریف شود که این کاربر در بیشتر اوقات، تنها دسترسی db\_dataread و db\_datawrite را نیاز دارد و در اختیار قراردادن نقش‌هایی مانند db\_owner یا sysadmin به این کاربر، تهدیدات زیادی را موجب می‌شود. زیرا این کاربر تنها باید عملیات DML را انجام دهد و عملیات DDL، مانند تغییر جداول، باید در تغییر مدل‌ها و با فایل اسکریپت که قبلاً در محیط تست، آزمون شده‌اند، توسط DBA انجام شود.

- با استفاده از راه‌اندازها، می‌توان در هنگامی که کاربری به پایگاه داده متصل می‌شود، می‌توان IP کاربر، نرم‌افزار کاربر و اطلاعات دیگر را به دست آورد. به کمک این راه‌اندازها، می‌توان کاربر برنامه کاربردی را محدود کرد که تنها از طریق برنامه به پایگاه داده متصل شود. به این ترتیب، کاربری نمی‌تواند از طریق SQL Studio Management به پایگاه داده متصل شده و داده‌های پایگاه داده را مشاهده کند.

- بیشتر تهدیدات به کمک راه‌کارهای فوق برطرف می‌شود اما نصب فایروال پایگاه داده یا Database Firewall می‌تواند از تهدیدات رایج دیگر جلوگیری کند.

- کد نرم‌افزار اهمیت بسزایی دارد. به نحوی که نرم‌افزار دارای آسیب‌پذیری مانند SQL Injection نباشد.

- در صورت امکان، بر روی سرور پایگاه داده، هیچ جزء اضافه‌ای نصب نشود، حتی کامپوننت‌های ویندوز. بهتر است بر روی سرور تنها سیستم عامل خام و پایگاه داده نصب شود.

- نصب patch‌های سیستم عامل و پایگاه داده و به‌روزرسانی، بسیار توصیه می‌شود.

- دسترسی‌های اضافه به سیستم عامل لغو شود.

## خلاصه‌ای از روش‌های مناسب کلی برای امنیت SQL Server

- لاگین‌های SQL Server به صورت منظم بازرسی شود.
- امتیازات محدود به SQL Server Service Account
- استفاده از گذرواژه‌های قوی و پیچیده برای حساب کاربری sa و لاگین‌های SQL Server
- اعمال خط مشی‌های گذرواژه و انقضای گذرواژه برای لاگین‌های SQL Server
- خودداری از استفاده از تصدیق اصالت SQL Server و استفاده از تصدیق اصالت ویندوز
- مرور دوره‌ای لاگین‌های ویندوز و SQL Server
- رمزنگاری نسخه‌های پشتیبان پایگاه داده SQL Server
- اجرای SQL Server در درگاه‌های متفاوت از درگاه‌های پیش‌فرض
- غیرفعال کردن SQL Server Browser Service
- مخفی‌سازی نمونه SQL Server
- رمزنگاری اتصالات به SQL Server
- غیرفعال کردن همه ویژگی‌های بلااستفاده SQL Server

## چک لیست امنیت SQL Server

### امنیت فیزیکی

- محدود کردن تعداد کارمندانی که به سخت افزار فیزیکی دسترسی دارند
- ذخیره سازی نسخه پشتیبان در یک محل امن
- پیکربندی اطلاع ها برای هشدارهای سخت افزاری

### امنیت سیستم عامل

- نصب همه service pack و آپدیت های ویندوز
- پیکربندی یک فایروال
- محدود کردن تعداد کارمندانی که دسترسی Windows Administrator را به SQL Server دارند

### نصب SQL Server

- نصب فقط اجزای مورد نیاز
- نصب همه service pack و آپدیت های مهم SQL Server
- غیرفعال کردن ویژگی ها و سرویس های غیرضروری
- غیرفعال کردن پروتکل های بلااستفاده SQL Server
- تغییر درگاه های پیش فرض SQL Server
- محدودسازی دسترسی به پیکربندی و فایل پایگاه داده SQL Server
- محدودسازی دسترسی به پوشه های نسخه پشتیبان SQL Server
- اجرای SQL Server Best Practice Analyzer برای راستی آزمایی هنگام نصب

- غیرفعال کردن گزینه xp\_cmdshell

## حساب‌های کاربری

- تغییر نام و غیرفعال کردن حساب کاربری sa، در صورتی که برنامه کاربردی اجازه می‌دهد.

- استفاده از حالت‌های Windows Authentication

- استفاده از حساب‌های کاربری سرویس برای برنامه‌های کاربردی

- پیکربندی حساب‌های کاربری سرویس با حداقل امتیازات

- امتیازات کاربر باید کمینه باشد.

- پیکربندی بازرسی لاگین SQL Server برای لاگین‌های موفق و ناموفق

[1] SQL Server 2012 Security Best Practices - Operational and Administrative Tasks, SQL Server White Paper, Bob Beauchemin, January 2012.

[[http://download.microsoft.com/download/8/f/a/8fabacd7-803e-40fc-adf8-355e7d218f4c/sql\\_server\\_2012\\_security\\_best\\_practice\\_whitepaper\\_apr2012.docx](http://download.microsoft.com/download/8/f/a/8fabacd7-803e-40fc-adf8-355e7d218f4c/sql_server_2012_security_best_practice_whitepaper_apr2012.docx)]

[2] Microsoft SQL Server Tutorial, [https://technet.microsoft.com/en-us/library/mt590198\(v=sql.1\).aspx](https://technet.microsoft.com/en-us/library/mt590198(v=sql.1).aspx)

[3] [www.greensql.net](http://www.greensql.net)

[۴] مصاحبه با مهندس مسعود نیکوفر، ادمین پایگاه داده بانک اقتصاد نوین