به نام خدا

امنیت بانک های اطلاعاتی

(رشته امنیت اطلاعات)

Data base Security

ویژه دانشجویان

تدوین : علی ثاقب موفق

پیشگفتار اهمیت امنیت پایگاه داده ها :

هدف و محور در امنیت اطلاعات، اطلاعات است. بنابراین تمام تلاش های ما برای امن سازی اطلاعات است. حال این اطلاعات در انحاء و شیوه های مختلف ذخیره و نگهداری می شوند و یکی از رایج ترین و موثرترین روش های ساماندهی و نظم دهی به اطلاعات استفاده از بانک های اطلاعاتی برای ذخیره سازی آنها است.

اطلاعات قابل استناد و با ارزش سازمانها و شرکت ها عموما در قالب برنامه های مکانیزه در بانک های اطلاعاتی ساماندهی می شوند تا امکان بازیابی موثر آنها فراهم گردد. با توجه به این نکات متوجه می شویم که مقصد اطلاعات، بانک اطلاعاتی است. لذا امنیت بانک اطلاعاتی از اهمیت ویژه ای برخوردار است و بایستی به آن پرداخته شود. در اینجا ضمن اشاره کوتاه به مفاهیم اصلی پایگاه داده ها و روابط آن به آسیب پذیری و تهدیدات آن می پردازیم و روش های امن سازی آن را بررسی می نماییم.

بانک اطلاعاتی رابطه ای:

بانک های اطلاعاتی که از جداول برای ذخیره اطلاعات استفاده می کنند را بانک اطلاعاتی رابطه ای گویند و به هر جدول یک رابطه گویند.

موجودیت (Object) : هرچیزی که بخواهیم در مورد آن اطلاعاتی ذخیره کنیم. موجودیت ها در بانک اطلاعاتی به صورت جدول ذخیره می شوند و هر موجودیت مستقل یک جدول (Table) است.

فیلد(Field) : فیلد به هر مشخصه ای می گویند که برای نگهداری اطلاعات در جدول ها استفاده می کنیم. به عبارتی دیگر کوچکترین واحد داده ذخیره شده در بانک اطلاعاتی، فیلد است. صفت های(attribute) هر موجودیت همان فیلد جدول هستند. مانند نام، نام خانوادگی، شماره ملی و تاریخ تولد

رکورد(Record): مجموعه ای از فیلدهای اطلاعاتی مرتبط به هم و ذخیره شده درباره یک نمونه از شیء را رکورد گویند که همان سطر ها در جداول می باشند.(Row)

کلید(Key) : کلید یک فیلد اطلاعاتی از جدول است که می تواند یک رکورد خاص را معرفی نماید و به عبارتی کلیدها راه شناسایی رکوردها در جداول هستند. کلید، صفت یا ستونی است که برای هر کدام از رکوردها مقدار منحصر به فردی دارد . کاربرد دیگر کلید، استفاده از آنها برای ایجاد ارتباط بین جداول است.

انواع کلید:

کلید کاندید(Candidate Key): هر فیلد یا ترکیبی از فیلدها که با استفاده از آن بتوان رکوردهای یک جدول اطلاعاتی را متمایز نمود مثل کد ملی که متمایز کننده رکوردهای جدول اطلاعاتی است یا ترکیب دو فیلد نام + کد ملی که باز هم متمایز کننده خواهد بود. اینها کلید کاندید هستند. در یک جدول تعداد زیادی کلید کاندید هست. یعنی در یک جدول اطلاعاتی برای کلید شدن کاندیدهای زیادی داریم.

کلید اصلی(Primary Key): کلید کاندیدی که هیچ قسمت کوچکتر و ساده تر آن خودش دیگر کلید نباشد، کلید اصلی است. مثلا " نام+ کد ملی" را نمی توان کلید اصلی نامید چون کد ملی خودش به تنهایی متمایز کننده است. ولی کلید کاندید، "کد ملی" ، کلید اصلی است. اگر در یک جدول چند کلید اصلی وجود داشته باشد. کلید اصلی ساده تر و کوتاهتر را به عنوان کلید اصلی انتخاب می کنیم.

کلید خارجی(Foreign Key): اگر در جدول "ب" فیلدی داشته باشیم که آن فیلد در جدول "الف" کلید اصلی باشد. به آن فیلد در جدول "ب" کلید خارجی می گویند. فرق کلید اصلی و خارجی این است که مقادیر کلید اصلی در جدول تکرار نمی شود ولی مقادیر کلید خارجی در جدول تکرار می شوند.

مثل بانک اطلاعات دانشجویان که شامل جدول اطلاعات دانشجو؛ شامل فیلدهای شماره دانشجویی، نام، نام خانوادگی، رشته تحصیلی است و جدول دروس ترم دانشجو شامل: شماره دانشجو، شماره درس و کد استاد است که در آن شماره دانشجو تکرار می شود. لذا شماره دانشجو در جدول اطلاعاتی اول کلید اصلی و در جدول اطلاعاتی دوم این شماره دانشجویی به تعداد دروس انتخاب شده توسط دانشجو تکرار می شود و کلید خارجی است.

تعریف و ایجاد یک رابطه (Relation): برای ایجاد یک رابطه، کلید اصلی از یک جدول اطلاعاتی را با کلید خارجی از جدول دیگر ارتباط می دهیم. بنابراین برای ایجاد یک ارتباط بین دو جدول اطلاعاتی به دو کلید نیاز خواهیم داشت.

انواع رابطه (Relation):

رابطه یک به یک: حالتی است که در آن هر رکورد از جدولی مانند "الف" با یک رکورد از جدول "ب" در ارتباط باشد. مثل اطلاعات شناسنامه ای دانشجو شامل شماره دانشجویی و اطلاعات فارغ التحصیلان دانشگاه که شامل شماره دانشجویی نیز می شود.

رابطه یک به چند: در این رابطه هر رکورد از جدول "الف" با چند رکورد از جدول "ب"ارتباط دارد. از نگاه جدول "ب" رابطه چند به یک است. مثل رابطه درس و گروه درسی و استاد و گروه درسی، به طوریکه یک درس ممکن است به چند گروه درسی تعلق داشته باشد و همچنین یک استاد در چندین گروه درسی تدریس نماید. در اینجا کلید اصلی جدول "الف" به عنوان کلید خارجی به جدول "ب" افزوده می شود.

رابطه چند به چند: رابطه ای که در آن چند رکورد از جدول " الف" با چند رکورد ازجدول "ب" ارتباط دارند. در این رابطه باید از یک جدول واسط استفاده کرد که کلید اصلی هر دو تا جدول را دارد. مثل رابطه بین اساتید و دروس. در این حالت ما به اطلاعات هر دو جدول دسترسی خواهیم داشت. مثل اینکه از کد استاد در جدول اطلاعات اساتید به اطلاعات دروس مورد تدریس همان استاد دسترسی خواهیم داشت. چند استاد می تواند چند درس و چند درس می تواند توسط چند استاد، تدریس شود. در این حالت یک جدول جدید ساخته می شود که شامل کلیدهای هر دو طرف است.

نرمال سازی(Normalization)وکاهش افزونگی اطلاعات: یعنی اطلاعات فقط در یک مکان (جدول) ذخیره می شوند نه این که چند بار در جداول مختلف تکرار شوند. اطلاعات در تمام بانک ها با استفاده از ایجاد رابطه بین جداول و relationship قابل دسترس باشند.

حفظ یکپارچگی اطلاعات(Integrity): اعمال تغییرات روی اطلاعات در یک مکان و جدول انجام شده ولی آثار تغییرات آن در تمام جداول و بانک های اطلاعاتی مشاهده شود.

۱_ بانک اطلاعاتی امن (Secure Data Base **):**

هدف از امن سازی پایگاه داده هادر واقع حفاظت از اطلاعات است که مدیریت مجوزها و سطوح دسترسی کاربران را به همراه داردو برای این منظور طراحی پایگاه داده باید شامل؛ تحلیل، طراحی مفهومی، طراحی تفضیلی، پیاده سازی، تست و نگهداری باشد و برای امن سازی پایگاه داده بایستی مکانیزم های امنیتی در دو سطح DBMS و OS اعمال شود. این دو سطح دارای تفاوت هایی می باشند:

- دانه بندی: درجه دانه بندی درOS در سطح فایل است درحالیکه در DBMS ریزتر و در سطح رابطه، سطر و فیلد است.
- وجود رابطه معنایی میان داده ها در DBMS
- در OS اشیاء فیزیکی مثل فایل ها وجود دارند ولی در DBMS اشیاء منطقی مثل View نیز داریم.
- در DBMS چند نوع داده و چند نوع دسترسی (مثل دسترسی مدیریتی و آماری) وجود دارد ولی در OS سه مد دسترسی X,R,W داریم.

مکانیزم های امنیتی که DBMS باید تامین نماید:

- درجات مختلف دانه بندی برای دسترسی (رابطه، تاپلها و پایگاه داده)
- حالات مختلف دسترسی، کنترل دسترسی های مختلف و تمایز R , W
- انواع مختلف کنترل دسترسی برای یک درخواست دسترسی
  - ✓ وابسته به اسم اشیاء
  - ✓ وابسته به داده
  - ✓ وابسته به زمینه(وابسته به زمان و مکان)
  - ✓ وابسته به تاریخچه

- ✓ وابسته به نتیجه(نتایج پرس جو ها)
- ✓ حداقل مجوز
- ✓ تفکیک وظایف

۱-۱_انواع مدل های کنترل دسترسی:

- اجباری(mandatory): در این حالت، هر شیئ یک برچسب امنیتی دارد و کاربران نیز نیز دسته بندی می شوند و سپس کاربران متعلق به هر دسته متناسب با طبقه بندی شغلی خود به مجوعه شیئ هایی که دارای برچسب خاص مربوطه هستند، منتسب می شوند.
- اختیاری (Discretionary): در این حالت، کاربر به نحوه دسترسی داده های خود کنترل دارد و از برچسب امنیتی استفاده نمی شود. هر شیئ یک لیست کنترل دسترسی دارد که شامل فهرست کاربران مجاز است.به طوریکه کاربران دارای دسترسی به هر شیئ مشخص می باشد.
- مبتنی بر نقش(role base): با توجه به اینکه هر کارمند دارای نقشی خاص در سازمان می باشد. لذا در این حالت ارائه حق دسترسی صرفا مبتنی بر عملیاتی است که کاربر در سازمان می تواند متناسب با شغل خود انجام دهد و لذا مجوزها به نقش های تعریف شده کاربران در سازمان اختصاص داده می شود. مثل نقش حسابداری و یک کاربر در صورتی مجوز انجام عملی را دارد که قبلا نقشی به آن انتصاب شده باشد.
- کنترل دسترسی مبتنی بر قوانین(Rule Based - RBAC): دسترسی بر اساس یک سری قوانین که توسط مدیریت تعریف شده است ارائه می شود و برای هر شیی لیست کنترل دسترسی(ACL) تهیه می گردد که درآن قوانین لازم برای داشتن حق دسترسی آورده شده است. مثلا یک قانون اینکه کاربر یا گروهی از کاربران تنها در روزهای خاص و در ساعات خاص به شبکه وصل شوند.

دستورات ارائه مجوز دسترسی:

- دستور GRANT در SQL: GRANT Rights ON TABLE TO user_list

- دستور REVOKE در SQL : REVOKE Rights ON TABLE FROM user_list

۲-۱_ نیازهای امنیتی پایگاه داده:

- شناسایی و تصدیق اصالت کاربر(Authentication)
- کنترل دستیابی(Access Control) ،دسترسی فقط به داده هایی که مجوز آنرا دارد. (Authorization)

۳-۱_ صحت پایگاه داده:

- صحت فیزیکی پایگاه داده: در مقابل قطع برق و مواردی مشابه که دسترسی پذیری پایگاه داده را مختل می سازد. پایگاه داده بایستی همواره در دسترس و پابرجا باشد. به طوریکه اطلاعات بانک اطلاعاتی همواره توسط کاربران مجاز قابل دسترسی باشد.(Availability)
- صحت منطقی پایگاه داده : ساختار پایگاه داده حفظ شود که مستلزم طراحی مناسب بانک اطلاعاتی است. بایستی فیلدهای جدول متناسب با مجوز کاربران برای آنها نمایش داده شود.
- صحت عناصر: داده ها به طور صحیح در عناصر ذخیره شود. داده ها بایستی صحیح، بروز، دقیق و صریح باشد. داده ها از منابع اصلی و درست گرفته شود و در صورت ورود داده ها توسط کاربر مراحل بررسی هوشمند داده ها توسط نرم افزار مربوطه صورت گیرد.

۴-۱_ قابلیت ردیابی و بازرسی:

ردیابی کاربرانی که به بانک اطلاعاتی دستیابی داشته و یا آنرا تغییر داده اند. همان مفهوم Log فایل ها و Auditing می باشد تا عملکرد کاربران بر روی بانک اطلاعاتی در فایل و محلی ذخیره و قابل پیگیری باشد و برای این منظور می توان از نرم افزارهای تحلیل log نیز استفاده نمود. در این حالت می توان متوجه شد که کاربر به چه جداول اطلاعاتی دسترسی داشته و اینکه کدام رکورد و فیلدها را دسترسی و رویت و یا عملی

روی آن انجام داده است و چنانچه تغییری در یک فیلد اطلاعاتی بوجود آمده باشد قابل پیگیـری و ردیـابی خواهد بود.

۵–۱_ دسترس پذیری (Availability): اطلاعات مجاز همواره در دسترس کاربران مجاز باشد.

۶–۱_ محرمانگی اطلاعات(Confidentiality):

دسترسی فقط توسط افراد مجاز صورت گیرد و فرد غیر مجاز به هیچ عنوان دسترسی به اطلاعـات جـداول بانک اطلاعاتی نداشته باشد. هر کاربر صرفا متناسب با مجوز های تعریف شده برای او مـی توانـد بـه همـان بخش های اطلاعات و یا همان فیلدها و جداول اطلاعاتی دسترسی داشته باشد.

۷–۱_ صحت یا سازگاری اطلاعات(Integrity) :

اینکه اطلاعات وسط راه تغییر نکند. عدم تغییر ناخواسته اطلاعات توسط افراد غیر مجاز بـه طوریکـه هـیچ فردی نتواند اطلاعات را دریافت و پس از اعمال تغییرات خود خواسته مجددا آنـرا بـرای کـاربر مجـاز دیگـر ارسال نماید. یعنی اطلاعات مذکور در جداول مبداء و مقصد یکی باشد. نام دیگر اختلال در اطلاعـات بـین دو کاربر مجاز Man In the middle است.

۲_ مساله استنتاج :

موضوعی است که در آن کاربر از طریق اطلاعات بدست آمده از پایگاه داده و با آگاهی های خارجی خود یـا نتایج پرس جو های قبلی و ترکیب آنها می تواند اطلاعات دیگری را استنتاج می نماید. یعنی اطلاعات مجاز، کاربری را قادر می سازد که چیزی درباره اطلاعاتی که مجاز نیست، استنتاج کند.

۱–۲_ کانال استنتاج :

استنتاج به معنی نتیجه گیری است و این موضوع برای داده های مرتبط با هم بسیار معمول است. اسـتنتاج در بانک اطلاعاتی وسیله یا روشی است که کاربر با استفاده از داده های طبقه بندی شده در سطح پایین که

مجاز به دسترسی به آنها است بتواند داده ای را از سطح بالاتر نتیجه گیری و استنتاج کند که مجاز به دسترسی به آن نیست. و این نقض امنیت اطلاعات است که بایستی نسبت به رفع ان اقدام نمود.

۲-۲_ روشهای متدوال استنتاج:

پرس جوی مستقیم روی داده های خاص: در این حالت، مستقیما داده حساس مورد سئوال قرار می گیرد و یا در شرایط پرس جو شرطی بر روی مقدار فیلد حساس گذاشته می شود و با توجه به نتیجه حاصل شده از پرس جو می توان در مورد مقادیر مربوطه اظهار نظر کرد. مثل پرس جوی کارکنانی که پاداش بیش از یک رقم مشخص دریافت نموده اند یا نمره دانشجویانی که کمتر از ۱۰ می باشد.

۲-۳_ استنتاج در پایگاه های داده آماری:

گاهی استفاده از آمار و بررسی تعداد داده ها و مقایسه آنها با همدیگرمی تواند ما را به نتایج مطلوب برساند. با جمع آوری آمارهای مختلف و کنار هم گذاشتن آنها می توان به مقادیر دلخواه رسید. یعنی آمارهایی را از پایگاه داده درخواست کرده و با جمع بندی آنها بتوان مقدار فیلد خاصی که دسترسی به آن ممکن نیست را دسترسی پیدا نمود. مثلا از تاریخ تولد به سن فرد دسترسی داشت و یا از فیلد تعداد فرزند به تاهل و تجرد فرد پی برد.

۲-۴_ ازطریق ترکیب داده و ابر داده:

دراین حالت، نتیجه گیری از داده ها با اطلاعات موجود خودمان و سایر داده های پیرامونی می باشد و با توجه به اینکه قوانین جامعیت معمولا از قوانین موجود در محیط نشات می گیرد و بیانگر قانون حاکم بین موجودیت ها و صفات محیط عملیاتی است، کاربران می توانند از آنها مطلع باشند. قوانین جامعیت کلید اصلی مشخص است. لذا کاربر می تواند با بدست آوردن یک سری اطلاعات سطح پایین مجاز و با استفاده از ابر داده ها به داده های خاص و سطح بالاتر دسترسی یابد. این درحالی است که کاربر مذکور ممکن است مجوز دسترسی به داده های مذکور نداشته باشد.

۲–۵_ قانون جامعیت کلید اصلی:

قانون جامعیت کلید اصلی یک نوع راه نفوذ است. هر رکورد در یک رابطه مقدار منحصر به فردی برای کلید اصلی دارد و نمی تواند مقدار تهی یا NULL به خود بگیرد. همین موضوع باعث می شود که قانون جامعیت کلید اصلی یک راه نفوذ باشد. حال با توجه به این موضوع ممکن است استنتاج هایی صورت گیرد .

فرض کنید رکوردی با مقدار کلید مشابه آنچه در جدول وجود دارد توسط کاربر سطح پایین درج شود. اگر پایگاه داده جلوی ورود این اطلاعات را بگیرد. معلوم می شود که چنین رکوردی قبلا در پایگاه داده وجود داشته است. که در این صورت یک نشتی اطلاعات خواهد بود.

حال اگر اجازه دهد چنین رکوردی درج شود . قانون جامعیت نقض شده است. یک راهکار برای آن چند نمونه سازی است. در این روش چندیدن رکورد همزمان در جدول با یک مقدار کلید درج می شود ولی هر کاربر رکورد مربوط به خودش را می بیند. بعنوان مثال برای درک اینکه یک دانشجو در بانک اطلاعاتی هست یا خیر؟ می توان شماره دانشجویی مذکور را در بانک درج نمود و منتظر پاسخ سیستم درخصوص درج دانشجوی جدید ماند و در صورت عدم قبول درخواست یعنی اینکه دانشجوی مذکور وجود دارد و ما به اطلاعات و هدف مورد نظر خود رسیده ایم.

۲–۶_ وابستگی تابعی (وابستگی چند مقداره):

فیلدهای داده معمولا با همدیگر ارتباط و وابستگی هایی نیز دارد . مثل محل تولد و داشتن خوابگاه و غیره . با توجه به وجود وابستگی هایی بین صفات خاص، استنتاج صورت می گیرد. فرض کنید که می دانیم تمامی افراد در رده کاری یکسان در یک شرکت، حقوق مشخصی را دریافت می نمایند. چنانچه فیلد حقوق محرمانه و فیلد رده کاری غیر محرمانه باشد. در این حالت با انجام پرس جو بر روی فیلد رده کاری جدول اطلاعاتی مربوطه می توان حقوق فرد را استنتاج نمود.

برای حل آن می توان کاری کرد که هرگاه چند فیلد با همدیگر وابستگی تابعی داشتند. سطح امنیتی آنها را با بالاترین سطح موجود بین آنها قرار داد تا امکان کشف اطلاعات فیلد حساس از طریق سایر فیلدهای داده میسر نباشد.

۲-۷_ قوانین محدودیتی مقداری استنتاج:

مثل سن افراد و یا مدت تحصیل دانشجو در دانشگاه که از طریق ترکیب اطلاعات سطح پایین پایگاه داده و علم به اینکه روی مقدار یک یا چند فیلد قانون محدودیتی از لحاظ مقدار وجود دارد. صورت می گیرد.

۲-۸_ محدودیت دسته بندی یا افراز:

دسته بندی انواع اطلاعات و اختصاص جداول مختلف به آنها منجر به افراز اطلاعات می شود . ممکن است اطلاعاتی در مورد افراز مقادیر یک فیلد وجود داشته باشد. مثلا فرض کنید که می دانیم هر فرد یا دانشجو است یا کارمند یا استاد. لذا دانستن یک مقدار ممکن است به استنتاج در مورد سایر مقادیر منجر شود.

۲-۹_ حملاتی مختلف به منظور استنتاج:

حمله های ردیاب : ردیابی داده ها یکی از شیوه های موثر در کشف اطلاعات است. مواقعی تعداد کمی از ورودی، بخش اعظم داده ای را منتشر می کند. یک حمله ردیاب می تواند مدیر پایگاه داده را فریب دهد و پرسش هایی اضافی که نتایج کوچک تولید می کند را انجام دهد تا مدیر پایگاه داده، اطلاعات مطلوب را در اختیار وی بگذارد. ردیاب، رکوردهای اضافی را جهت بازیابی برای دو پرسش مختلف اضافه می کند. دو مجموعه رکورد یکدیگر را حذف کرده و تنها آماری را که مطلوب است بجا می گذارد. مثلا لیست دانشجویان نخبه و لیست دانشجویان دارای خوابگاه و استفاده از فصل مشترک آنها.

۲-۱۰_ آسیب پذیری در مقابل دستگاه خطی:

در این حالت چندین پرس جو از بانک اطلاعاتی صورت گرفته و سپس نقاط مشترک بدست می آید. این حمله می تواند در حالت های غیر از حمله های آماری برای استنتاج بکار رود. با استفاده از جبر مجموعه ها

و با حل کردن دستگاه معادلات مجموعه ای خطی می توان به نتایج واحد دست یافت. در این حالت چندین پرسش از پایگاه داده صورت می گیرد و دستگاه معادلاتی تشکیل می شود به طوری که با حـذف بعضـی از شرایط بتوان به یک مقدار واحد که دسترسی به آن غیر مجاز بوده ، رسید.

۱۱-۲_ جلوگیری از استنتاج توسط محافظ ها:

معیارهایی برای ارزیابی محافظ های امنیتی در بانک اطلاعاتی موجود می باشد که عبارتند از:

- امنیت : به سطح محافظت در مقابل افشاء کلی یا جزء
- توانمندی: با فرض آگاهی نفوذ کننده به اطلاعات کمکی
- مناسب برای فیلدهای عددی و قطعی
- مناسب برای محافظت از بیش از یک فیلد محرمانه
- غنی بودن اطلاعات آشکار شده
- هزینه، پیاده سازی روش های کنترل امنیت و آموزش کاربران

۱۲-۲_ روش های مقیدکردن پرس جو:

- کنترل اندازه مجموعه پاسخ به پرسش
- بازرسی یا مرور: در این روش، تاریخچه ای از تمام پـرس جـو هـای انجـام شـده توسـط هـر کـاربر نگهداری شده و در موقع هر پرس جوی جدید آنهـا را جهـت افشـای احتمـالی اطلاعـات محرمانـه آزمایش می کند. مزیت این روش این است که مقید ساختن یا اختلال لازم نیست مگر آنکه افشاء امکان پذیر شود و عیب آن این است که احتمال تبانی بین کاربران وجود دارد.
- تقسیم بندی: مقادیر فیلدهای خاص و حساس گروه بندی می شوند و پرس جو در داخل گروه هـا انجام می گیرد تا از عدم افشاء آنها اطمینان حاصل شود.

- کتمان و پنهان سازی سلول: کتمان سلول برای جدول های منتشر شده از داده های اماری سرشماری شده بکار می رود. این روش تمام سلول های جدول را که شامل آمارهای حساس هستند را پنهان می سازد.

۳– روش های اختلال :

۳–۱_ جابجا کردن داده:

این روش، یک پایگاه داده را با پایگاه داده دیگر که دارای رکوردهایی متفاوت ولی آمارهای درجه آنها یکسان و ساختار یکسانی دارد، عوض می کند. یعنی ساختار یکی است ولی محتوی متفاوت است.

۳–۲_ گرد کردن:

به طرف بالا یا پایین تا نزدیکترین مضرب یک مبنای خاص, سه نوع گرد کردن داریم که شامل نظام یافته، تصادفی و کنترل شده است. در کنترل شده ؛ چندین سلول از یک جدول را طوری تغییر می دهد که جمع های واقعی ردیف ها و ستون ها حفظ شوند و تغییر در نتایج محاسبات حاصل نشود. در نظام یافته : گرایش خاصی ایجاد می کند و می توان بر آن فائق آمد. در تصادفی؛ مقدار آنرا به مقدار کمی تغییر می دهد. به طوریکه قابل توجه و پیگیری نباشد.

۴– پرس جو های نمونه گیری تصادفی:

این حالت، از مجموعه جواب یک پرسش بدست می آید. این روش از اینکه یک جاسوس بتواند ترکیبی از مجموعه جواب، پرسش را کنترل کند، جلوگیری می کند. اگر یک کاربر چندین بار یک پرسش مشخص که مجموعه جواب یکسان دارد را انجام دهد برای هر مرتبه یک نمونه تصادفی متفاوت به کار برده می شود و هر کاربر می تواند میانگین نتیجه را بدست آورد. و برای محافظت از این حفره امنیتی می توان معادل بودن این پرسش ها را کشف نمود و برای همه آنها نمونه یکسانی بکار برده شود. با اینکار مهاجم نمی تواند نتیجه گیری خاصی از پرسش خود بنماید.

۴–۱_ پرس جو های نمونه گیری تصادفی همراه با کنترل مجموعه پاسخ به پرسش:

پرس جوی نمونه گیری تصادفی در مقابل حمله هایی که اندازه مجموعه پاسخ آنها کوچک است آسیب پذیر می باشند. اما اگر در کنترل با هم ترکیب شوند. امنیت لازم بدست می آید.

۴–۲_ کشف کانال ها بعد از طراحی با ابزارهای خودکار:

در این روش سیستم بعد از طراحی توسط ابزارهای خودکار مورد بررسی قرار می گیرد و کانال هـای بـالقوه استنتاج کشف می گردند.

۵– نقطه آسیب پذیر Ms Sql :

سرویس دهنده مایکروسافت دارای چندین نقطه آسیب پذیر جدی است که امکان دریافت اطلاعات حساس و تغییر در محتویات بانک اطلاعاتی و به مخاطره انداختن حیات سرویس توسط مهاجمان را فراهم می کند. در برخی موارد و به دلیل عدم پیکربندی صحیح سیستم، میزبانان سرویس دهنده ممکن است در معرض تهدید و آسیب قرار گیرند. مثلا دو کرمslammer/sql.hell/sapphire sql snake / spida از نقاط ضعف شناخته شد Ms Sql استفاده کرده و توانستند در مدت زمان کوتاهی حملات گسترده ای را انجام دهند. همچنین بر اساس گزارش Internet Storm Center پورت های ۱۴۳۳ و ۱۴۳۴ Ms Sql پیش فرض، از جمله پورت هایی می باشند که توسط مهاجمین دائما مورد بررسی قرار می گیرد.

۵–۱_ مراحل کنترل دسترسی و تعیین اعتبار:

- تغییر پسورد های پیش فرض، فورا بعد از نصب دیتا پیش فرض
- قفل و LOCK کردن Account کاربرانی که در حال استفاده نیستند و حذف آنها در صورتی که اصلا از سیستم و بانک اطلاعاتی استفاده نمی کنند.
- الزام و enforce کردن کاربران برای استفاده از پسورد قوی مخصوصا کاربرانی که در سطح domail level تعیین اعتبار شده اند.

- برداشتن public account ها مثل دسترسی عمومی از همه account ها
- انتخاب اینکه یا domain authentication و یا database authentication داشته باشیم. و نه هردو آنها که موقع بروز مشکل، سردرگمی پیش آید.
- نقش ها و گروه ها را بدقت ازمایش کنید. مجوزهای کاربران را لیست کنید و از نقش ها و گروه ها و مجوزهای کافی آنها اطمینان حاصل کنید. این کار وقت گیر است ولی ضروری است و نقشه مجوزهای دسترسی صادره برای بانک اطلاعاتی باید تهیه شود.
- پشتیبانی و Protect کردن از عملیات admin از کاربران.
- تقسیم کردن وظایف admin برای شرکت هایی با بیش از یک ناظر بانک اطلاعاتی لازم است. وظایف ناظر بین admin ها تقسیم می شود.
- پلت فرم Plat Form بانک اطلاعاتی رابطه ای تمهیدات لازم برای کنترل دسترسی پیشرفته را دارد.

۵-۲_ ارزیابی پیکربندی بانک اطلاعاتی:

- متوجه شوید که بانک اطلاعاتی چطور تنظیم و پیکربندی شده است و آنرا با استاندارد آن بررسی نمایید.
- ماجول ها و سرویس هایی را که نیاز ندارید، بردارید.
- مستندسازی و پیکربندی پایگاه داده تایید شده باشد و اینکار بعدها بعنوان مرجع شما خواهد بود و برای همه admin ها قابل استفاده خواهد بود.

۵-۳_ ارزیابی تعاملی پلت فرم بانک اطلاعاتی:

- همه بانک های اطلاعاتی مستقیما با سیستم عامل ارتباط دارند.
- غیرفعال نمودن روال های ذخیره شده خارجی و توسعه یافته غیر ضرور
- اطمینان از اینکه admin دامین همان admin بانک اطلاعاتی نیست.
- حفظ محرمانگی مالک بانک اطلاعاتی محلی، ابزارها و startup script و وارده های رجیستری

۶- ارتباطات امن:

- شما باید مطمئن شوید که ارتباطات بانک اطلاعاتی شخصی حفظ می شود.
- رمزگذاری session ها بین برنامه کاربردی و بانک اطلاعات مخصوصا اتصالات web application
- ری ست کردن شماره پورت های بانک اطلاعاتی به مقدار غیر پیش فرض برای مثال انتقال پورت های پیش فرض ۱۵۲۱ اوراکل به مقدار تصادفی
- بلوک block کردن اتصالات ad-hoc . اتصالات ad-hoc از موقعیت های غیر مطلوب زمان، روز و یا برنامه کاربردی تایید نشده کشف وتوسط فایروال دیتابیس وسیستم کنترل دسترسی رد می شود.

۷- بروز نگهداری نرم افزاری پایگاه داده Patch the Database **:**

- ایجاد یک فرایند و شرایط انجام database patch
- اجازه ندهید که patch توسط dba خاص بارگزاری شود.
- sysnکردن patch با فروشنده Vendor
- پیکربندی مجدد کردن در حالتی که patch قابل پذیرش نیست.

۸- مرور رویداد log **:**

- استفاده از logging
- ایجاد یک سیاست بازیابی log
- تعیین رویدادهایی که فیلتر کردن آنها نیاز نیست.
- -مرور دوره ای log ها و تمرکز روی fail ها و login ها
- -مرور تنظیمات log دوره ای

۹- احاطه کردن امنیت embrace insecurity**:**

- انبار و inventory بانک اطلاعاتی
- کشف کاتالوگ کردن داده های حساس

- داشتن طرح روی آنچه انجام می دهید وقتی داده از بین می رود و یا دزدیده می شود.
- داشتن طرح پوشش خرابی disaster recovery plan
- ایجاد یک فرهنگ مشارکتی و همکرای

۱۰– مونیتورینگ فعالیت دیتا بیس:

آنچه معمولا DAM نامیده می شود درواقع مونیتورینگ فعالیتها است. آنها تمام فعالیتهای sql در بانک اطلاعاتی را گرفته که شامل اعمال نظارتی و تحلیل حملات و استفاده بد از امنیت است و رفتار مهاجمان را تحلیل می کنند.

این ابزار می تواند یک تنوع وسیع از تهدیدها و قابلیت بلوک کردن حملات معین را کشف کند و می تواند مسیر و رد رویه ای را برای ارائه گزارش منظم جمع آوری نماید.

۱۱– تزریق SQL یا SQL Injection :

حمله ای است که در آن مهاجم می تواند کد مخرب را اجراء نماید (Malicious Code) که به واسطه آن سرور بانک اطلاعاتی مربوط به یک برنامه کاربردی WEB را کنترل کند و این یکی از قدیمی ترین و خطرناک ترین آسیب پذیری های WEB Application است.

با Sql Injection کاربر می تواند مکانیزم های تعیین هویت و اعتبار را دور بزند و محتوی کامل بانک اطلاعاتی را بازیابی کند. این حمله می تواند منجر به اضافه کردن، حذف رکورد و در نهایت جامعیت داده ها را مختل کند.

نحوه عمل Sql Injection: به منظور اجرای کد مخرب Sql یک مهاجم نیاز دارد که ابتدا یک ورودی در برنامه کاربردی Web داخل یک sql query پیدا کند. وب سایت آسیب پذیر شامل ورودی مستقیم کاربر در یک حمله SQL است. مهاجم سپس یک payload را درج می کند که بعنوان بخشی از sql query شامل

خواهد شد و در مقابل سرور بانک اطلاعاتی آنرا اجراء می کند. بعنوان مثال سمت سرور کد زیر برای تعیین هویت کاربر در برنامه کاربردی وب استفاده می شود:

```
#define post variables
Uname = request.post[‘username’]
Password = request.post[‘password’]
#sql query vulnerable to sql
Sql = “SELECT id FROM  users WHERE  username = ‘ “ + uname + “ ‘ AND password = ‘  “ + passwd = “ ‘ “
#execute the sql statement
Database.execute(sql)
```

کد script بالا, مثال ساده از تعیین هویت کاربر با username ,password در برابر بانک اطلاعاتی با جدول users و ستون username , password است. مثال ساده یک تزریق sql می تواند چیزی به سادگی تنظیم فیلد پسورد به password ‘or 1=1 باشد. که در آن Sql query نتیجه زیر در برابر بانک اطلاعاتی اجراء می شود.

```
SELECT id FROM  users  WHERE  username = ‘username’ AND  password = ‘password’  or 1=1’
```

مهاجم می تواند در حمله sql ،کنترل اجرای query را داشته باشد. وقتی که پرس جو اجرا می شود نتیجه به برنامه برگشت داده می شود و نتیجه آن دور زدن تعیین هویت کاربر خواهد بود.

۱۲– سرریز بافر Buffer Overflow :

وقتی یک برنامه یا پروسس سعی درنوشتن داده بیشتری در یک بلوک حافظه یا بافر با طول ثابت حافظه ای دارد که بافر به خود اختصاص داده است و داده اضافی بر روی آدرسها ی مجاور بافر مقصد بازنویسی می شوند. اگر داده اصلی شامل اشاره گر بازگشت توابع باشد و آدرسی است که پروسس بایستی سپس برود . در این حالت یک مهاجم می تواند برای اشاره به آدرس این انتخاب مقدار جدیدی را set کند. مهاجم معمولا

مقادیر جدید را به اشاره گر به محلی که payload قرار گرفته است را set می کند و کنترل را به کد مخرب مهاجم انتقال می دهد.

زبانهای برنامه نویسی C++ , C الزامی به حمایت برنامه نویس از نوشتن داده های اضافی ندارند. زبان برنامه نویسی C امکان نوشتن در هر قسمت از حافظه و دستکاری مستقیم حافظه را میسر میسازد. زبانهای مدرن perl , java , C# شانس خطای کمتری را در کد نویسی می دهند که اسیب پذیری buffer overflow است. اما buffer overflow در جاییکه دستکاری مستقیم حافظه اجازه داده می شود. ممکن است در هر شرایط برنامه نویسی رخ دهد .

دو نوع Buffer Overflow وجود دارد. به نام های heap based , stack based . پشته یا stack فضای پشت سر هم حافظه است که برای سازماندهی داده ها، همراه با فراخوانی تابع است. شامل پارامترهای تابع، متغیرهای محلی تابع و اطلاعات مدیریت مثل frame و اشاره گرهای دستور العمل. فضای حافظه Heap ساختار حافظه ای است که برای مدیریت پویای حافظه استفاده می شود. برنامه نویسان اغلب برای تخصیص حافظه ای که سایز آن در زمان کامپایل ناشناخته است از آن استفاده می کنند.

رایج ترین دلیل موضوع حمله های buffer over flow این است که برنامه کاربردی، تخصیص حافظه و تعیین اعتبار ورودی client یا پروسس های دیگر را به خوبی مدیریت نمی کند.

دستورات مرتبط با امنیت بانک اطلاعاتی

Grant Privilage ON Object TO user

Privilage: select, insert, update, delete, reference, alter, all

Reference : توانایی ایجاد یک **constrain** جهت ارجاع به جدول

Object: نام جدول

User: نام کاربر یا public

GRANT select, insert, update, delete ON employee TO smith

REVOKE privilages ON object FROM user1

For example: REVOKE DELETE ON employee from Anderson

برای ایجاد یک کاربر، ابتدا باید در پایگاه داده login ایجاد کنیم:

Create login [domain-name/login-name] from windows [with default-database = database-name | default-language = language-name];

در sql server:

Create login login-name with password = { ‘password’ | hashed-password HASHED} [MUST-Change] [, sid = sid-value] | default-database = database-name | default-language = language-name]

CREATE primary key
Create table table-name
(
Column1 datatype [NULL | not NULL] [primary key]
Column2 datatype [NULL | Not NULL],
……
);

DROP primary key
Alter table table-name

DROP CONSTRAIN constrain-name;
CREATE Table employees
(
Last-name varchar(50) NOT NULL,
First-name varchar(50) Not NULL,
Salary MONEY;
CONSTRAINT employees-PK PRIMARY KEY (last-name, first-name)
);

ALTER TABLE table-name ADD CONSTRAINT constraint-name PRIMERY KEY (column1 , column2 ,…);

ALTER Table table-name
DROP CONSTRAINT constraint-name;

ALTER LOGIN login-name WITH password = ‘password’

Grant update(ename) ON emp to xyz
Grant update(column-name) on table-name to user-name
Grant Select On emp to xyz with grant option;

Grant update(ename) , insert(empno,ename) on emp to xyz;

تنها اختیارات insert, update, reference می تواند روی سطح ستون مجوز داده شود.

Grant update(column-name) on tablename to username;

نقش ها، مجموعه ای از اختیارات هستند. وقتی تعداد کاربران زیادی در بانک اطلاعاتی هستند، ارائه مجوز و یا اخذ آن مشکل است. بنابراین اگر نقش ها تعریف شوند، می توان مجوز را به نقش ها داد. برخی از مجوزهای مربوط به نقش های سیستم عبارتند از :

CONNECT create table, create view, create synonyms, create sequence, create session, ….
RESOURCE create procedure, create sequence, create table, create trigger, …
DBA All system Privilages

نحو دستور syntax:

Create ROLE role-name[identified by password]
مثال : Create ROLE testing [identified by pwd];

مثال: برای ارائه مجوز به create table به کار می رود توسط نقش:

Create ROLE testing
Grant create table to testing;
Grant testing to user1;

REVOKE Create Table From testing;
DROP ROLE role-name;

حذف نقش از یک بانک اطلاعاتی

DROP ROLE testing;

```
GRANT   SELECT TO [myuser];
DENY    SELECT TO [myuser];
REVOKE SELECT TO [myuser];

Grant   SELECT ON dbo.employee( employee ID, firstname, middlename,
surname) to HR_intern
Grant       SELECT(name) ON mydb, user to 'mysqluser'@mysqlHOST'
Grant  SELECT(col1), INSERT( col1, col2) ON mydb.mytb1 TO
'someuser'@somehost';
Grant       Select On emp to xyz with grant option;
```

ارائه مجوز به emp برای کاربر xyz و اینکه او قادر باشد این اجازه را به کسان دیگر بدهد.

ایجاد نقش:

```
Create Role testing;
Grant Create Table     To     testing;
Grant testing To user1;
REVOKE Create Table From testing;
Drop Role role-name;
Drop Role testing;
```

Login ID :

برقراری ارتباط، بخشی از اطلاعات اعتبار کاربر است که که کاربر برای دستیابی به Sql Server ارائه می دهد. بعد از بررسی درستی ID برقراری ارتباط توسط SQL سرور، تعیین اینکه کاربر مجاز به دستیابی سرویس دهنده است، کاربر باید بانک اطلاعاتی را انتخاب نماید و بعد ID برقراری ارتباط را با تمام ID های کاربری آن بانک مقایسه نماید تا اطمینان حاصل شود که کاربر به بانک اطلاعاتی دسترسی دارد.

فرایند بررسی اعتبار :

وقتی کاربری اقدام به دستیابی به سرویس دهنده می کند چهار چیز کنترل می‌شود. در هر مقطع کنترل می شود که کاربران مجاز به دستیابی هستند یا خیر . اگر چنین باشد امکان پیشروی به آنها داده می شود در غیر این صورت کاربر،پیام خطایی دریافت خواهد کرد وکار وی متوقف می شود.

نخستین سطح این امنیت، در **سطح شبکه است** کاربران در بیشتر مواقع با یک شبکه ویندوز NT ارتباط برقرار خواهند کرد. اما با هر شبکه دیگری که در کنار شبکه مذکور وجود داشته باشد نیز می توانند

ارتباط برقرار کنند. کاربر باید یک IDابرقراری ارتباط و یک کلمه عبور معتبر شبکه وارد کند. در غیراین صورت پیشرفت کار وی در همین سطح متوقف خواهد شد. به نظر می رسد که چند روش برای گذشتن از این سطح امنیتی وجود دارد. اما واقعا این طور نیست. مواقعی است که یک کاربر بخواهد با استفاده از یک بانک اطلاعاتی واکشی کند اگر چه کاربر برای دستیابی به شبکه ممکن است نیاز به وارد کردن اطلاعات برقراری ارتباط نداشته باشد اما account ی که برنامه کاربردی در سرویس دهنده تحت آن درحال اجرا است حتما باید با شبکه ارتباط برقرار کند.

دومین سطح امنیت در خود **سرویس دهنده** است وقتی کاربر به این سطح می‌رسد ، می بایست یک ID برقراری ارتباط و یک کلمه عبور معتبر برای پیشروی ارائه کند بسته به مد امنیتی که در سرویس دهنده به کار می برید.

سومین‌سطح امنیت نیاز به یک کاربری ID در بانک اطلاعاتی است که می خواهد به آن دستیابی داشته باشد چنانچه کاربری، ID در بانک اطلاعاتی نداشته باشد، کار زیادی نمی تواند انجام دهد مگر اینکه یک IDکاربری مهمان در یک بانک اطلاعاتی وجود داشته باشد در این حالت آنچه که رخ می دهد آن است که کاربر از طریقID برقراری ارتباط به سرویس دهنده دستیابی پیدا کرده است. اما به بانک اطلاعاتی دستیابی ندارد و بانک اطلاعاتی حاوی یک IDکاربری مهمان است. مجوزها را می‌توان به کاربری ID مهمان تخصیص داد درست به همان گونه ای که به کاربران دیگر قابل تخصیص هستند بانک های اطلاعاتی جدید طبق پیش فرض فاقدID کاربری مهمان هستند.

آخرین سطح امنیت SQL ، ارتباط امنیتی با مجوز ها است. سرور با مجوزهایی سرو کار دارد. آنچه که در این سطح رخ می دهد آن است کهSQL سرور کنترل می کند که ID کاربری که کاربر از طریق آن به سرویس دهنده دستیابی پیدا کرده است مجوزهای دستیابی به شی ء های مورد نظر را داشته باشد. این امکان وجود دارد که دستیابی تنها برای برخی از شی ء های آن بانک اطلاعاتی باشد ونه تمام شی ء ها که معمولا نیز اینگونه است. سطوح امنیت می توانند دستیابی به سرویس دهنده را برای کاربران غیر مسئول بسیار دشوار کنند.

نقش ها:

بجای تخصیص مجوز به هر یک از کاربران بانک اطلاعاتی، به راحتی می توانید یک نقش ایجاد کنید و مجوزها را به آن نقش تخصیص دهید.

نقش برنامه کاربردی: این نقش یک نقش ویژه است که امکان دستیابی به داده های موجود در بانک اطلاعاتی را تنها به برنامه های کاربردی خاص می دهد و کاربران را مجاب می کند تا از طریق برنامه کاربردی به داده های بانک اطلاعاتی دستیابی پیدا کنند. این ویژگی از اهمیت خاصی برخوردار است. چرا که بیشتر کاربران قادر به استفاده از برنامه های کاربردی خاصی چون مایکروسافت اکسس خواهند بود که می تواند با بانک اطلاعاتی ارتباط برقرار کند و دستیابی مستقیم به جداول موجود در آن را برای کاربران فراهم کند. این کار کاربران را مجاب به استفاده از برنامه کاربردی می کند که خصوصا برای دستیابی به داده ها نوشته شده و از دستیابی برنامه های کاربردی دیگر جلوگیری می شود .

نقش ها، برای گروه بندی کاربرانی به کار می روند که نیازهای دستیابی آنها مشابه است و عبارتند از :

۱) نقش های از پیش تعریف شده سرویس دهنده

۲) نقش های از پیش تعریف شده بانک اطلاعاتی

۳) نقش عمومی

۴) نقش های شخصی بانک اطلاعاتی

نقش های از پیش تعریف شده سرویس دهنده :

با تعریف این نقش می توان بعضی از کارهای مدیریتی سرویس دهنده را به کاربران واگذار نمود. هفت نقش از پیش تعریف شده سرویس دهنده به شرح زیر هستند :

sysadmin: اعضای این نقش دارای اختیارات کامل بر روی سرویس دهنده هستند، مشابه Accountی به نام sa در SQL سرور است.

serveradmin: اعضای این نقش می توانند پیکربندی مشخصات سرویس دهنده را انجام دهند .

setupadmin: مجوز اضافه یا حذف پیوندها و ارتباطات سرویس دهنده

Securityadmin: مدیریت ارتباطات سرویس دهنده و امنیت مثل Grant , Revoke

processadmin: مدیریت پراسس ها و فرایندهای اجرایی SQLسرور و امکان قطع یک پراسس

dbcreator: مجوز ایجاد بانکهای اطلاعاتی در سرویس دهنده

diskadmin: مجوز ایجاد و مدیریت فایل ها در دیسک.

نقش های از پیش تعریف شده بانک اطلاعاتی : نقش های از پیش تعریف شده بانک های اطلاعاتی به کاربران امکان می دهند تا عملیات گوناگونی را انجام دهند و عبارتند از :

db-owner: اعضای این گروه به عنوان مالک بانک اطلاعاتی تعریف می شوند.

db-accessadmin: امکان اضافه یا حذف دسترسی کاربران در بانک اطلاعاتی

db-datareader: رویت تمام داده های جداول کاربری بانک اطلاعاتی

db-datawriter: مجوز افزودن، تغییریا حذف داده های تمام جداول کاربری بانک اطلاعاتی

db-ddladmin: مجوز اینکه شئ های یک بانک اطلاعاتی راحذف، اضافه یا اصلاح کند .

db-securityadmin: نقش ها و اعضای نقش های بانک اطلاعاتی SQLسرور را مدیریت کنند .

db-backupoperator: اعضای این نقش مجوز تهیه نسخه پشتیبان از بانک اطلاعاتی را دارند .

db-denydatareader: اعضای این گروه نمی توانند هیچیک از داده های بانک اطلاعاتی را ببینند .

db-denydatawriter : اعضای این گروه هیچ مجوزی برای تغییر داده های بانک اطلاعاتی ندارند.

نقش عمومی

نقش عمومی، یک نوع نقش بانک اطلاعاتی ویژه است که تمام کاربران بانک اطلاعاتی عضوی از آن هستند و به هنگام ایجاد تمام بانک های اطلاعاتی ایجاد می شود فایده این نقش زمانی مشخص می شود که می خواهید مجموعه ای از مجوزهای پیش فرض را به تمام کاربران بدهید.

نقش های شخصی بانک اطلاعاتی

نقش هایی هستند که مدیر سیستم برای مقاصد ویژه ایجاد می کند. این نقش ها توانایی تخصیص مجوزهای ویژه را فراهم می کنند که در نقش های از پیش تعریف شده موجود نیستند. قوانینی که می بایست به هنگام ایجاد این نقش ها به خاطر داشته باشید، عبارتند از :

نقش های شخصی بانک اطلاعاتی در محدوده یک بانک اطلاعاتی ایجاد می شوند و در وسط چند بانک اطلاعاتی قابل گسترش نیستند. کاربران در هر لحظه نمی توانند به بیش از یک نقش شخصی بانک اطلاعاتی تعلق داشته باشند. نقش های شخصی می توانند حاوی IDهای برقراری ارتباط ویندوزNT، ID های برقراری ارتباطSQL سرور و دیگر نقش هایSQL سرور باشند

نقش های سیستم یا system role :

CONNECT : Create table, create view, create synonyms, create sequence,…

RESOURCE : Create Procedure, Create Sequence, Create Table,….

DBA : All system Privilage

DDL: Data Defination Language → create

DML: Data ManipulationLanguage →update

TCL: Transaction Control Language →Rollback, savepoint

DCL: Data Control Language →Grant, Revoke

دستور alter table در SQL : برای اضافه کردن یا حذف یک ستون کامل از جدول

Alter Table table-name ADD column-name datatype

Alter Table table-name DROP column-name datatype

دستور DELETE برای حذف اطلاعات یک رکورد در جدول است.

DELETE FROM table-name WHERE some_column = some_value

DELETE FROM PERSON WHERE ID= “16”

دستور * DELETE حذف همه رکورد های جدول

WAF – Web Application Firewall

این نوع فایروال یک لایه امنیتی بین مشتریان و برنامه یا سرویس فراهم می کند. یک WAF تمامی دسترسی ها به برنامه وب را فیلتر می کند و با امن کردن ساختار برنامه و همچنین کاربر برنامه، مکمل فایروال در شبکه می باشد و لایه دیگری از امنیت را به ارمغان می‌آورد. برنامه ها و نرم افزارهای وبی می توانند با تهدیدات زیادی آسیب پذیر شوند که به وسیله فایروال ها در شبکه قابل تشخیص نمی باشند. این نوع حملات می تواند بسیار شدید باشد. رایج ترین حملات عبارتند از:

تزریق کد :حملات SQL Injection از فرمهای وبی یا دیگر مکانیزم ها برای ارسال دستورات SQL یا دستورات شامل کاراکتر های خاص SQL استفاده می کنند. با ارسال این دستوراتSQL ، حمله کنندگان می توانند بخش مدیریت پایگاه داده ی SQL را مورد هدف قرار داده و دستورات تزریق شده را اجرا کنند و کاربران غیر مجاز به اطلاعات حساس پایگاه داده دسترسی پیدا کنند.

حملات Cross-Site Scripting یا XSS به دلیل عدم اعتبار سنجی ورودی های کاربران می باشد و مهاجمان می توانند با تزریق اسکریپت های مخرب در سایت از این آسیب پذیری سوء استفاده کنند.
افشای اطلاعات حساس و محرمانه: اگر برنامه ی وب شما از اطلاعات محرمانه وحساس مانند شماره کارت اعتباری یا اعداد اجتماعی (SSN. Social Security Number) محافظت نکند، حمله کنندگان قادر به سرقت اطلاعات هویتی و محرمانه، کلاه برداری و دیگر جرایم می شوند.

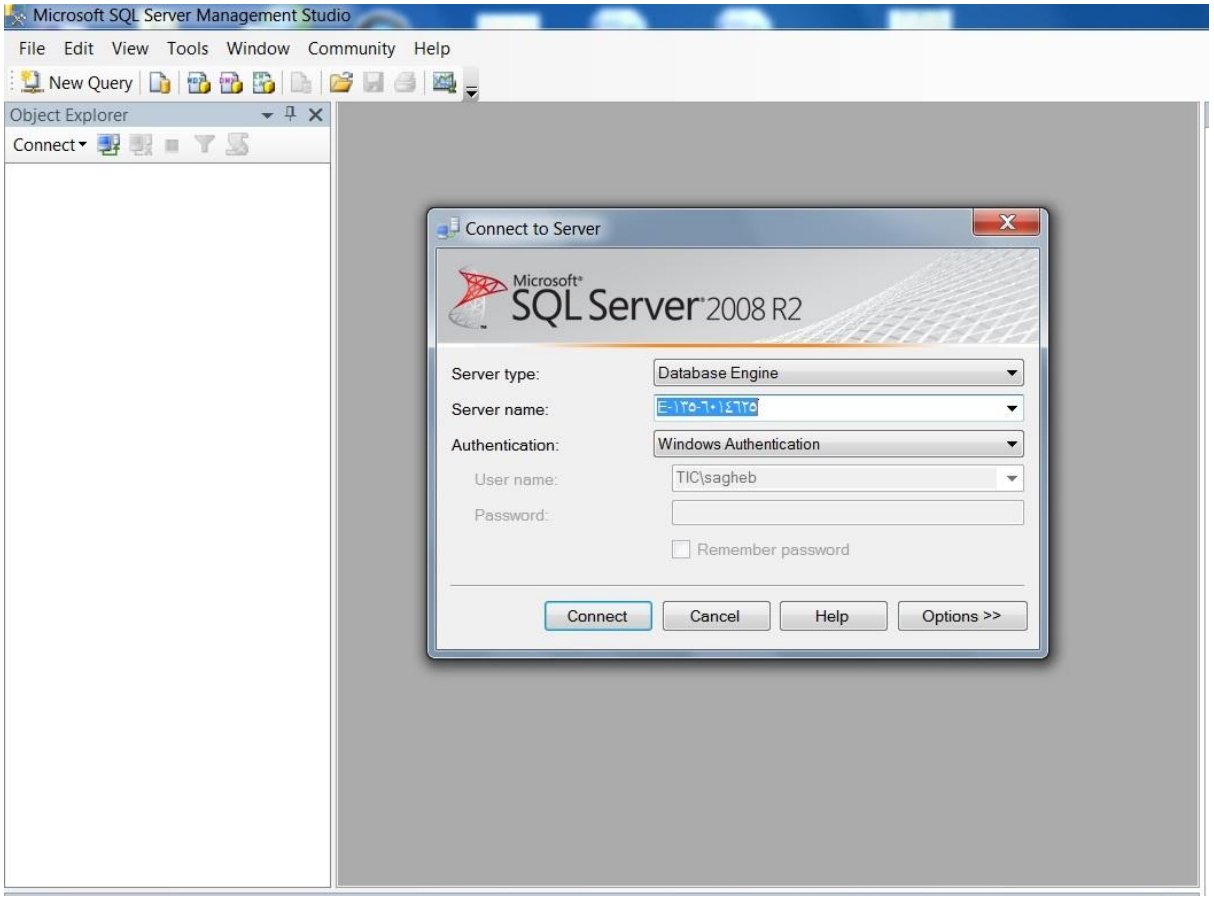
جعل تقاضای عبور سایتCSRF - Cross-Site Request Forgery :

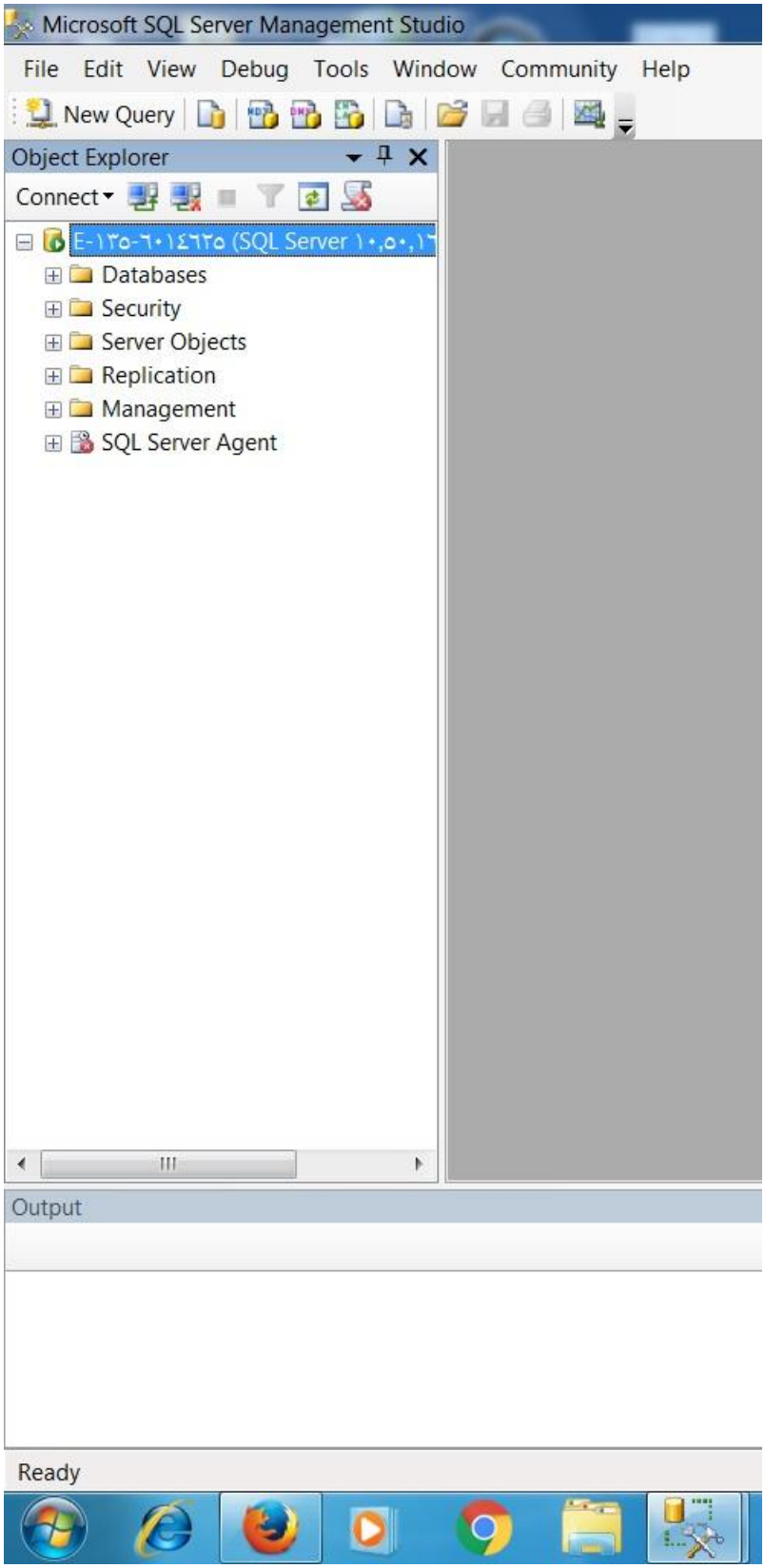
حملات CSRF کاربر را مجبور به ارسال درخواست HTTP که شامل کوکی های نشست قربانی است به برنامه وب آسیب پذیر می کند، برای برنامه وب آسیب پذیر این درخواست که از طرف قربانی می‌آید، قانونی و مجاز به نظر می رسد. WAF یک دیوار حفاظتی کامل با مکانیزم های امنیتی برای حفاظت از

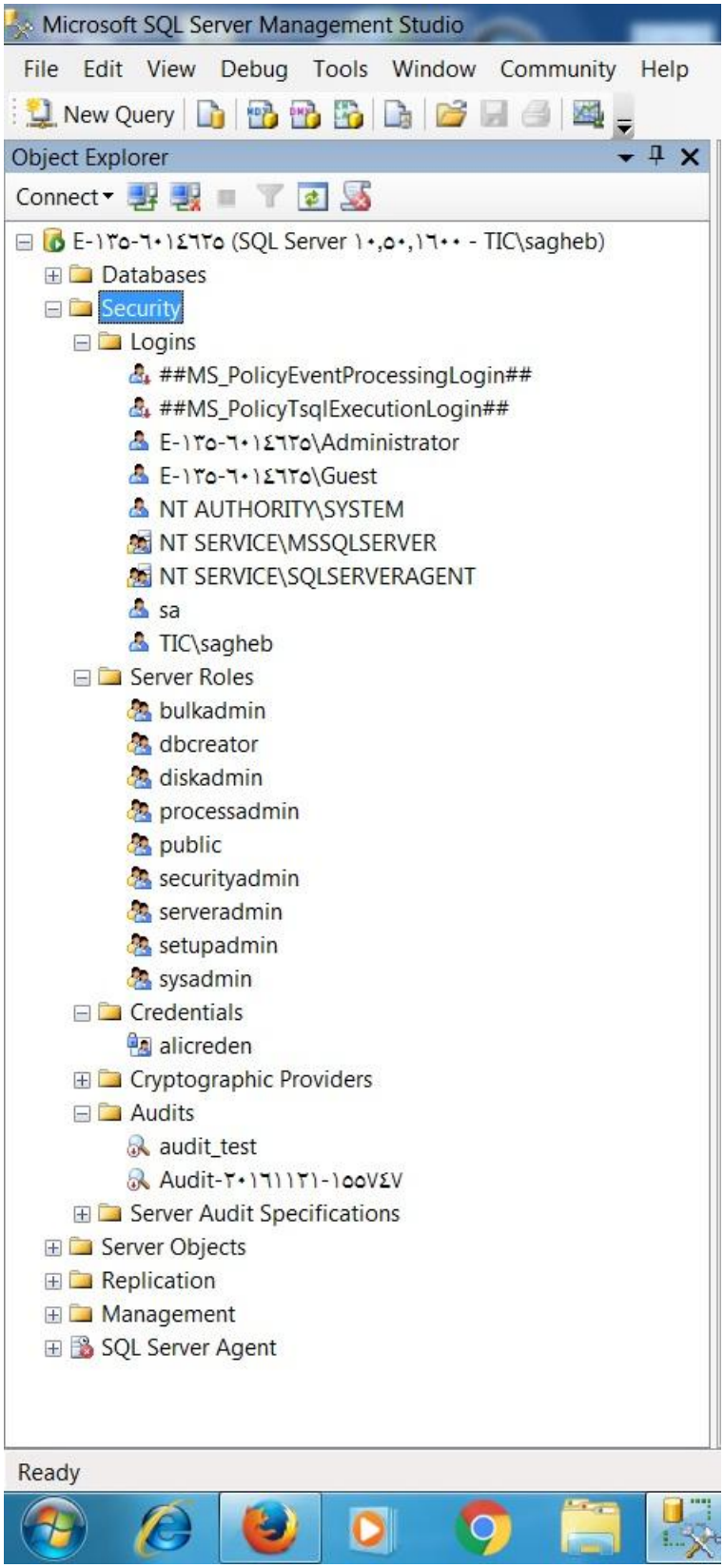
برنامه ی وب را فعال می کند که تضمینی در مقابل آسیب پذیری کد ها و جلوگیری از نشست اطلاعات می باشد.

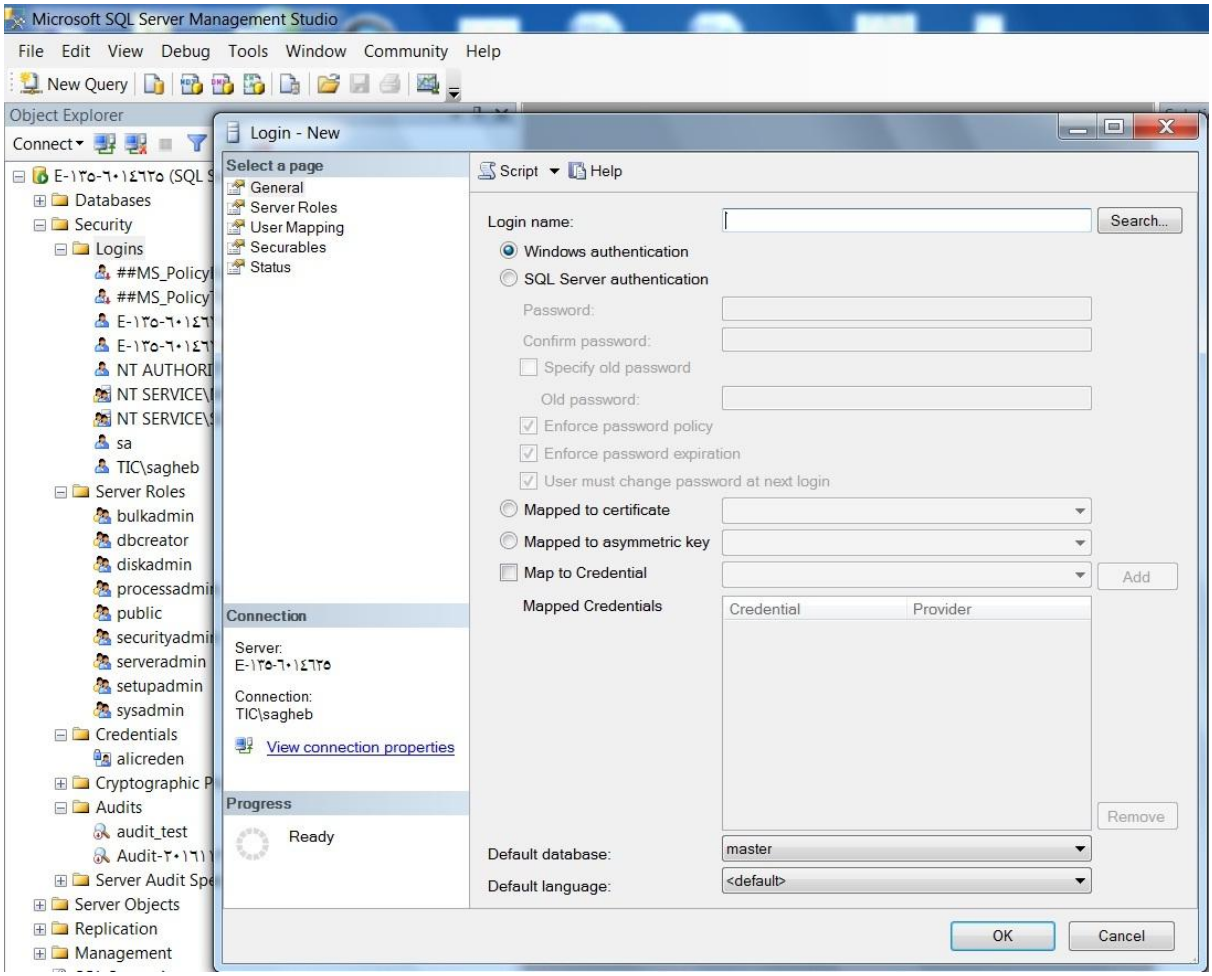
یک WAF نباید تنها محدود به ارایه حفاظت در لحظه ی خاص باشد. برای تأمین امنیت کامل و درست و تأثیر گذار برنامه های وب یک WAF باید بتواند با کنترل شناسایی کاربران نیز همراه باشد. همچنین نیاز به راهکار پیاده سازی ساده مدیریت دسترسی به وب (Web Access Management) نیز وجود دارد.

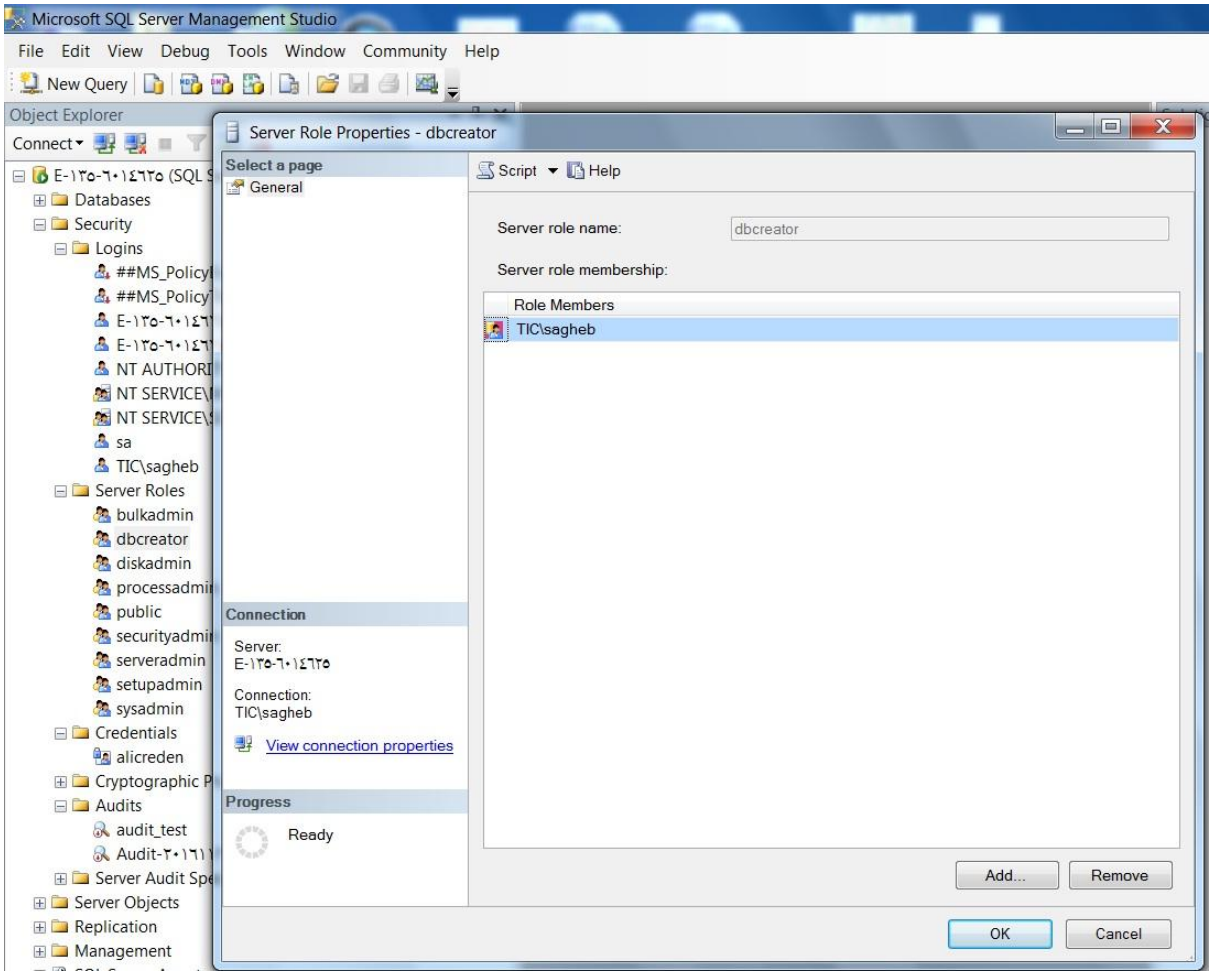
تصاویر مربوط به نرم افزار بانک اطلاعاتی SQL و نقش ها و امنیت در زیر آمده است:




Microsoft SQL Server Management Studio
File Edit View Debug Tools Window Community Help
New Query
Object Explorer
Connect
E-۱۳۵-۶۰۱٤٦۳٥ (SQL Server ۱۰.٥۰.۱٦
Databases
Security
Server Objects
Replication
Management
SQL Server Agent
Output
Ready


Microsoft SQL Server Management Studio
File Edit View Debug Tools Window Community Help
New Query
Object Explorer
Connect
E-١٣٥-٦٠١٤٦٢٥ (SQL Server ١٠,٥٠,١٦٠٠ - TIC\sagheb)
Databases
Security
Logins
##MS_PolicyEventProcessingLogin##
##MS_PolicyTsqlExecutionLogin##
E-١٣٥-٦٠١٤٦٢٥\Administrator
E-١٣٥-٦٠١٤٦٢٥\Guest
NT AUTHORITY\SYSTEM
NT SERVICE\MSSQLSERVER
NT SERVICE\SQLSERVERAGENT
sa
TIC\sagheb
Server Roles
bulkadmin
dbcreator
diskadmin
processadmin
public
securityadmin
serveradmin
setupadmin
sysadmin
Credentials
alicreden
Cryptographic Providers
Audits
audit_test
Audit-٢٠١٦١١٢١-١٥٥٧٤٧
Server Audit Specifications
Server Objects
Replication
Management
SQL Server Agent
Ready


Microsoft SQL Server Management Studio
File Edit View Debug Tools Window Community Help
New Query
Object Explorer
Connect
Databases
Security
Logins
Server Roles
bulkadmin
dbcreator
diskadmin
public
serveradmin
setupadmin
sysadmin
Credentials
alicreden
Audits
audit_test
Server Objects
Replication
Management
Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
Search...
Windows authentication
SQL Server authentication
Password:
Confirm password:
Specify old password
Old password:
Enforce password policy
Enforce password expiration
User must change password at next login
Mapped to certificate
Mapped to asymmetric key
Map to Credential
Add
Mapped Credentials
Credential
Provider
Remove
Default database:
master
Default language:
<default>
Connection
Server:
Connection:
TIC\sagheb
View connection properties
Progress
Ready
OK
Cancel



من الله التوفیق