**دانشکده مهندسی کامپیوتر**

**پایان‌نامه کارشناسی**

**طراحی و پیاده‌سازی ابزار مقاوم سازی پایگاه داده SQL SERVER**

**نگارش:**

**ریحانه دهقان**

**استاد راهنما:**

**دکتر مرتضی امینی**

**دی ۱۴۰3**

****

# چکیده

یکی از سازکارهای امنیت در سامانه‌های نرم افزاری، مقاوم سازی نرم‌افزارهای پایه مورد استفاده در سامانه ها از جمله سیستم مدیریت پایگاه داده است. در این پایان‌نامه ابتدا تهدیدات امنیتی رایج مرتبط با سیستم مدیریت پایگاه داده SQL Server از جمله حملات تزریق SQL، دسترسی غیرمجاز و سوءاستفاده از تنظیمات پیش‌فرض مورد بررسی و مرور قرار گرفته اند. سپس استانداردهای CIS به‌عنوان یک چارچوب معتبر برای مقاوم سازی و بهبود امنیت پایگاه داده بررسی و روش‌های عملی پیاده‌سازی آن‌ها تشریح شده است. این استانداردها شامل مواردی همچون مدیریت دسترسی‌ها، رمزنگاری داده‌ها، مانیتورینگ فعالیت‌ها و پیکربندی امنیتی پیشرفته است.نتایج حاصل از پیاده‌سازی این ابزار روی یک پایگاه داده‌ی نمونه نشان داد که تعداد آسیب‌پذیری‌های موجود در SQL Server به‌طور قابل توجهی کاهش یافته و مقاومت سیستم در برابر تهدیدات افزایش پیدا کرده است. همچنین، ارزیابی اثربخشی اقدامات انجام‌شده نشان داد که استانداردهای CIS می‌توانند به عنوان رویکردی عملی و کارآمد در سازمان‌ها برای محافظت از داده‌های حساس مورد استفاده قرار گیرند.

کلیدواژه‌ها: امنیت پایگاه داده، SQL Server، مقاوم سازی ،استانداردهای CIS

فهرست مطالب

[چکیده 1](#_Toc188341582)

[فصل 1- مقدمه 4](#_Toc188341583)

[فصل 2- امنیت پایگاه داده 6](#_Toc188341584)

[1-2-مبانی و چالش‌ها 6](#_Toc188341585)

[2-2-رویکردهای اساسی برای امن‌سازی پایگاه داده‌‌ها 7](#_Toc188341586)

[3-2- استانداردهای امنیتی پایگاه‌داده‌ها 9](#_Toc188341587)

[فصل 3- سامانه‌های ارزیابی مخاطرات 11](#_Toc188341588)

[1-3- دسته‌بندی پیشنهادات امن‌سازی: 11](#_Toc188341589)

[1-1-3-نصب به‌روزرسانی‌ها و وصله‌ها 11](#_Toc188341590)

[2-1-3-کاهش سطح حمله 11](#_Toc188341591)

[3-1-3-احراز هویت و مجوزها 16](#_Toc188341592)

[4-1-3- سیاست‌های رمز عبور 18](#_Toc188341593)

[5-1-3- نظارت و ثبت گزارش‌ها 18](#_Toc188341594)

[6-1-3- توسعه نرم‌افزار 19](#_Toc188341595)

[7-1-3- رمزنگاری 19](#_Toc188341596)

[3-1-8- سایر ملاحظات 20](#_Toc188341597)

[فصل 4- معماری و پیاده‌سازی سامانه پیشنهادی 22](#_Toc188341598)

[1-4-معماری سامانه پیشنهادی 22](#_Toc188341599)

[2-4-پیاده‌سازی 23](#_Toc188341600)

[1-2-4-پیکر‌بندی 24](#_Toc188341601)

[2-2-4-توصیف کلاس‌ها 26](#_Toc188341602)

[3-4-ارزیابی نتایج 31](#_Toc188341603)

[فصل 5- جمع بندی و پیشنهادات 37](#_Toc188341604)

فهرست شکل‌ها

[شکل ‏4‑1: ارتباط میان اجزای مختلف سامانه 24](#_Toc187806494)

[شکل4- 2 : ConnectionString 28](#_Toc187806495)

[شکل 4-3:کلاس AdminControls 30](#_Toc187806496)

[شکل 4-4: فرم ورود 34](#_Toc187806497)

[شکل 4-5: نمای کلی برنامه 36](#_Toc187806498)

[شکل 4-6: وضعیت ارزیابی امنیتی قبل از اجرای امن‌سازی 37](#_Toc187806499)

[شکل 4-7: وضعیت ارزیابی بعد از اجرای امن‌سازی 37](#_Toc187806500)

فهرست ‌جداول

[جدول 1 : ساختار فایل config 26](#_Toc187082537)

[جدول 2 : طبقه بندی پرسمان‌های ارزیابی 27](#_Toc187082538)

[جدول 3 : طبقه بندی پرسمان‌های امن‌سازی 28](#_Toc187082539)

# مقدمه

در عصر حاضر، داده‌ها به عنوان یکی از ارزشمندترین دارایی‌های سازمان‌ها و کسب‌وکارها شناخته می‌شوند. حفاظت از این داده‌ها و تضمین امنیت آن‌ها، به یکی از اصلی‌ترین چالش‌ها در حوزه فناوری اطلاعات تبدیل شده است. پایگاه داده‌ SQL Server به عنوان یکی از پرکاربردترین پایگاه داده ها، نقش بسزایی در ذخیره، مدیریت و بازیابی داده‌ها دارد. با این حال، به دلیل ماهیت حساس اطلاعاتی که در این سیستم‌ها نگهداری می‌شود، همواره در معرض تهدیدات امنیتی قرار دارند.

با گسترش روزافزون حملات سایبری و تهدیدات داخلی، امنیت اطلاعات به یکی از دغدغه‌های اصلی مدیران فناوری اطلاعات تبدیل شده است. این تهدیدات ممکن است از منابع مختلفی مانند هکرهای خارجی، کاربران داخلی بدون مجوز، و یا بدافزارها ناشی شود. عواقب این تهدیدات شامل سرقت اطلاعات حساس، خرابکاری در داده‌ها، ایجاد اختلال در خدمات و حتی از دست دادن اعتبار سازمانی است. این چالش‌ها نیازمند استفاده از راهکارهای نوآورانه و استانداردهای پیشرفته امنیتی هستند تا سطح اطمینان و اعتماد به سیستم‌های اطلاعاتی ارتقاء یابد.

پایگاه داده SQL Server به دلیل ویژگی‌های منحصر به فرد و توانمندی‌های پیشرفته‌ای که ارائه می‌دهد، در بسیاری از سازمان‌ها و صنایع مختلف مورد استفاده قرار می‌گیرد. با این وجود، گستردگی قابلیت‌ها و پیچیدگی این سامانه‌ها می‌تواند منجر به وجود حفره‌های امنیتی و آسیب‌پذیری‌هایی شود که توسط مهاجمان مورد سوءاستفاده قرار گیرند. برای کاهش این مخاطرات، استانداردهای امنیتی نقش کلیدی ایفا می‌کنند. این استانداردها مجموعه‌ای از بهترین شیوه‌ها و دستورالعمل‌های کاربردی هستند که به سازمان‌ها در ایمن‌سازی سیستم‌های خود کمک می‌کنند.

استاندارد CIS (Center for Internet Security) ، یک چارچوب جامع امنیتی است که برای ارزیابی و مقاوم‌سازی سامانه‌های اطلاعاتی در برابر تهدیدات طراحی شده است. این استاندارد با ارائه فهرستی از تنظیمات پیشنهادی، سازمان‌ها را در بهبود امنیت و کاهش خطرات امنیتی یاری می‌دهد. این استاندارد در زمینه‌های مختلف مانند پیکربندی‌های امن، دسترسی‌ها، رمزنگاری، و نظارت امنیتی برای پایگاه‌های داده مختلف از جمله SQL Server، توصیه‌های مفیدی دارد.

هدف این پروژه، طراحی و پیاده‌سازی ابزاری کاربردی برای مقاوم‌سازی پایگاه داده SQL Server با استفاده از استاندارد CIS است. این ابزار نه تنها وضعیت امنیتی فعلی پایگاه داده را تحلیل می‌کند، بلکه تنظیمات امنیتی مورد نیاز را نیز پیشنهاد داده و امکان اعمال این تنظیمات را فراهم می‌سازد.

در ادامه این گزارش، ابتدا در فصل 2 مفاهیم بنیادی امنیت پایگاه داده و استانداردهای مطرح در این حوزه بررسی خواهد شد. سپس در فصل 3، استاندارد CIS و ویژگی‌های آن در حوزه پایگاه داده‌های SQL Server به تفصیل مورد مطالعه قرار می‌گیرد. سپس در فصل 4 با مرور کارهای پیشین، روش‌شناسی پیاده‌سازی ابزار پیشنهادی تشریح می‌شود و نتایج حاصل از پیاده‌سازی ابزار و تأثیرات آن بر امنیت پایگاه داده تحلیل خواهد شد. در نهایت در فصل 5 جمع‎بندی و سوی کارهای آتی ارائه خواهد شد.

# امنیت پایگاه داده

به هرگونه عیب یا ضعف در طراحی، پیاده‌سازی، عملکرد و مدیریت یک سیستم که مورد سوء استفاده قرار بگیرد و بتواند سیاست‌های امنیتی آن سیستم را نقض نماید، مخاطره یا آسیب‌پذیری گفته می‌شود.  
در ادامه‌ی این فصل، مبانی امنیت و مخاطرات رایج پایگاه داده را توضیح خواهیم داد، رویکردهای اساسی برای ایمن‌سازی پایگاه داده‌‌ها را شرح میدهیم سپس درباره استاندارد‌های موجود که برای شناسایی و ارزیابی مخاطرات پایگاه داده به کار می‌روند صحبت خواهیم کرد.

## 1-2-مبانی و چالش‌ها

امنیت پایگاه داده‌ به مجموعه‌ای از اقدامات و تکنیک‌ها اطلاق می‌شود که برای حفاظت از داده‌ها در برابر دسترسی‌های غیرمجاز، دستکاری، یا افشا، به‌کار می‌روند. از آنجا که پایگاه‌داده‌ها مرکز نگهداری اطلاعات حساس و حیاتی سازمان‌ها هستند، تأمین امنیت آن‌ها اهمیت زیادی دارد. امنیت داده‌ها نه تنها از نظر قانونی و اخلاقی برای سازمان‌ها حیاتی است بلکه از نظر اقتصادی نیز بسیار مهم است.

با افزایش تهدیدات سایبری و پیچیده‌تر شدن حملات، داشتن یک رویکرد مناسب برای تأمین امنیت پایگاه داده‌ها ضروری است. در این بخش، برخی از تهدیدات و آسیب‌پذیری‌های متداول در امنیت پایگاه داده‌ها بررسی می‌شوند:

حملات تزریق[[1]](#footnote-1) SQL: این نوع حمله زمانی رخ می‌دهد که مهاجم قادر باشد کد SQL را در ورودی‌های برنامه وارد کند و از آن برای دسترسی به داده‌ها یا تغییر آن‌ها استفاده کند. تزریقSQL یکی از شایع‌ترین و خطرناک‌ترین نوع حملات علیه پایگاه داده‌ها است.

دسترسی غیرمجاز: ممکن است برخی از کارکنان یا مهاجمان با استفاده از دسترسی‌های غیرمجاز به داده‌ها نفوذ کنند. یکی از چالش‌های امنیتی، تخصیص نادرست مجوزهای دسترسی است که می‌تواند به راحتی منجر به افشای اطلاعات حساس شود.

حملات منع سرویس[[2]](#footnote-2) و منع سرویس توزیع شده[[3]](#footnote-3): این حملات می‌توانند در دسترس‌پذیری پایگاه داده‌ها اختلال ایجاد کنند و عملکرد سیستم‌ها را دچار مشکل کنند.

عدم پیکربندی صحیح: پیکربندی نادرست پایگاه داده‌ها می‌تواند به راحتی آسیب‌پذیری‌های امنیتی ایجاد کند. از جمله این آسیب‌پذیری‌ها می‌توان به استفاده از رمز عبورهای ضعیف، عدم فعال‌سازی ویژگی‌های امنیتی مانند رمزنگاری، و باز بودن پورت‌های غیر ضروری اشاره کرد.

## 2-2-رویکردهای اساسی برای امن‌سازی پایگاه داده‌‌ها

ایمن‌سازی پایگاه داده‌ها شامل اقداماتی در سطوح مختلف از جمله پیکربندی صحیح، کنترل دسترسی، نظارت، و به‌کارگیری ابزارهای امنیتی برای شناسایی تهدیدات است. در اینجا برخی از رویکردهای اصلی برای ایمن‌سازی پایگاه داده‌ها معرفی شده است:

کنترل دسترسی

یکی از مهم‌ترین اقدامات در زمینه امنیت پایگاه داده، مدیریت صحیح دسترسی[[4]](#footnote-4) به اطلاعات است. استفاده از اصل حداقل امتیاز[[5]](#footnote-5) به این معنی است که کاربران باید تنها به منابعی دسترسی داشته باشند که برای انجام وظایف خود به آن نیاز دارند. احراز هویت چندعاملی[[6]](#footnote-6) برای تأمین امنیت بیشتر در سطح دسترسی کاربران می‌تواند به کاهش ریسک‌های امنیتی کمک کند.

رمزنگاری

رمزنگاری[[7]](#footnote-7) داده‌ها نه تنها در زمان انتقال داده‌ها از طریق شبکه (برای جلوگیری از شنود و حملات مرد میانی[[8]](#footnote-8)) بلکه در زمان ذخیره‌سازی نیز از اهمیت بالایی برخوردار است. رمزنگاری داده‌ها به گونه‌ای طراحی شده است که تنها افراد مجاز قادر به دسترسی به اطلاعات رمز شده باشند.

استفاده از رمزگداری داده‌های شفاف[[9]](#footnote-9) و Always Encrypted در SQL Server برای رمزنگاری داده‌ها، به محافظت از اطلاعات در برابر دسترسی‌های غیرمجاز کمک می‌کند.

پشتیبان‌گیری و بازیابی

برای جلوگیری از از دست دادن داده‌ها در شرایط بحرانی مانند حملات سایبری، فاجعه‌های طبیعی یا خطاهای انسانی، انجام پشتیبان‌گیری[[10]](#footnote-10) منظم از اطلاعات در پایگاه داده ضروری است.

استراتژی‌های بازیابی[[11]](#footnote-11) اطلاعات باید به گونه‌ای طراحی شوند که اطمینان حاصل شود که داده‌ها در صورت بروز مشکلات قابل بازیابی هستند.

نظارت و گزارش‌گیری

نظارت[[12]](#footnote-12) مستمر بر فعالیت‌های پایگاه داده و ثبت گزارش‌[[13]](#footnote-13)های دقیق از دسترسی‌ها و تغییرات داده‌ها، ابزاری حیاتی برای شناسایی تهدیدات و فعالیت‌های مشکوک است.

استفاده از ابزارهای نظارتی مانند SQL Server Profiler برای پیگیری دسترسی‌ها و عملکرد سیستم می‌تواند در شناسایی حملات بالقوه بسیار مفید باشد.

## 3-2- استانداردهای امنیتی پایگاه‌داده‌ها

استانداردهای امنیتی مجموعه‌ای از دستورالعمل‌ها و بهترین شیوه‌ها هستند که به سازمان‌ها کمک می‌کنند تا سیستم‌های خود را در برابر تهدیدات امنیتی محافظت کنند. این استانداردها برای ایمن‌سازی سیستم‌های اطلاعاتی، شامل پایگاه‌داده‌ها، طراحی و معرفی شده‌اند. در این بخش، به بررسی استانداردهای معروف در حوزه امنیت پایگاه داده‌ها پرداخته می‌شود.

استاندارد NIST

استاندارد موسسه ملی فناوری و استانداردها[[14]](#footnote-14)**،** NIST ، استانداردهای جامعی برای امنیت داده‌ها و سیستم‌های اطلاعاتی دارد که بر اساس آن‌ها، سازمان‌ها می‌توانند سیستم‌های خود را از لحاظ امنیتی بهبود دهند. به‌ویژه، NIST توصیه‌هایی برای رمزنگاری داده‌ها و حفاظت از اطلاعات حساس در پایگاه‌داده‌ها دارد.

استاندارد STIG

این استاندارد که توسط وزارت دفاع ایالات متحده منتشر شده است، دستورالعمل‌های امنیتی دقیق برای پیکربندی سیستم‌ها و نرم‌افزارها را فراهم می‌کند. STIG[[15]](#footnote-15)‌ها به‌طور خاص برای سیستم‌های مختلف مانند سیستم‌های عامل ، سرورها، برنامه‌های کاربردی، شبکه ها و دستگاه‌های ذخیره سازی داده، راهنمایی‌های امنیتی ارائه می‌دهد.

استاندارد CIS

مرکز امنیت اینترنت[[16]](#footnote-16)، CIS ، در جهت شناسایی، توسعه، اعتبارسنجی، ارتقا و پشتیبانی از بهترین راهکارهای عملی برای حفاطت سایبری فعالیت می‌کند. راهکار‌های این استاندارد در دو بخش اصلی ارائه می‌شود:

کنترل ها[[17]](#footnote-17): این بخش به بررسی کنترل‌های امنیتی اساسی می‌پردازد که برای حفاظت از سیستم‌های اطلاعاتی اساسی استفاده می‌شوند.

معیار ها [[18]](#footnote-18): در این بخش راهنمایی‌هایی برای تنظیمات امنیتی مختلف ارائه شده است که بر اساس بهترین رویکردهای صنعت تهیه شده‌اند.

در فصل 3 به معیارهای ارزیابی بر اساس استاندارد CIS میپردازیم.

# سامانه‌های ارزیابی مخاطرات

سامانه‌های ارزیابی مخاطرات یکی از ابزار‌هایی هستند که در حوزه امنیت کامپیوتر، کاربرد زیادی دارند و هدف اصلی آن‌ها جمع‌آوری اطلاعات درباره یک شبکه، ارزیابی سیستم‌های موجود در شبکه و شناسایی و گزارش کردن ضعف‌های امنیتی است. در این فصل، ابتدا ‌دسته‌بندی امن‌سازی ها بر اساس استاندارد CIS را بررسی می‌کنیم سپس فرآیند سنجش و موارد کاربرد ابزار‌های مقاوم سازی را بررسی می‌کنیم.

## 1-3- دسته‌بندی پیشنهادات امن‌سازی:

پیشنهادات امن‌سازی پایگاه داده SQL server 2019 طبق استاندارد CIS به هفت دسته کلی تقسیم می‌شود که در ادامه به اختصار هر یک را شرح میدهیم.

### 1-1-3-نصب به‌روزرسانی‌ها و وصله‌ها

این بخش به نصب صحیح SQL Server و اعمال به‌روزرسانی‌های امنیتی مربوط می‌شود:

۱.۱: اطمینان از نصب آخرین به‌روزرسانی‌ها و وصله[[19]](#footnote-19)‌های امنیتی SQL Server

توصیه می‌شود که همیشه از آخرین نسخه و به‌روزرسانی‌های SQL Server استفاده شود تا از آسیب‌پذیری‌های امنیتی جلوگیری شود.

۱.۲: استفاده از سرورهای تک‌منظوره برای SQL Server

SQL Server باید روی سرورهایی که تنها برای این منظور اختصاص داده شده‌اند نصب شود تا سطح حمله کاهش یابد و مدیریت راحت‌تر شود.

### 2-1-3-کاهش سطح حمله

۲.۱: غیرفعال کردن Ad Hoc Distributed Queries

این گزینه اجازه می‌دهد تا کاربران بتوانند به داده‌ها از منابع خارجی (مانند دیگر پایگاه‌های داده) با استفاده از پرس‌وجوهای توزیع‌شده ad hoc استفاده کنند. غیرفعال کردن این گزینه، امنیت را افزایش می‌دهد زیرا امکان سوءاستفاده از آن برای اجرای کدهای ناامن یا دسترسی غیرمجاز به داده‌های خارجی کاهش می‌یابد.

۲.۲: غیرفعال کردن CLR Enabled

CLR[[20]](#footnote-20) اجازه می‌دهد که اسمبلی‌های .NET در SQL Server اجرا شوند. با غیرفعال کردن این گزینه، حملات احتمالی از طریق کدهای CLR که ممکن است دارای آسیب‌پذیری‌های امنیتی باشند، متوقف می‌شود و سطح حمله کاهش می‌یابد.

۲.۳: غیرفعال کردن Cross DB Ownership Chaining

این ویژگی اجازه می‌دهد که کاربران با دسترسی به یک پایگاه داده، به اشیاء پایگاه داده‌های دیگر نیز دسترسی پیدا کنند. غیرفعال کردن آن باعث محدودیت دسترسی بین پایگاه‌های داده مختلف می‌شود و امنیت اطلاعات را از دسترسی غیرمجاز محافظت می‌کند.

۲.۴: غیرفعال کردن Database Mail XPs

قابلیت Database Mail XPs اجازه ارسال ایمیل از داخل SQL Server را می‌دهد. غیرفعال کردن آن کمک می‌کند تا از احتمال سوءاستفاده برای ارسال ایمیل‌های ناخواسته یا حتی داده‌های حساس جلوگیری شود و در عین حال احتمال حملات Denial of Service (DoS) کاهش یابد.

۲.۵: غیرفعال کردن Ole Automation Procedures

این ویژگی به SQL Server این امکان را می‌دهد که عملیات‌های خارجی از طریق OLE Automation را اجرا کند. غیرفعال کردن آن، خطر اجرای کدهای ناامن یا دسترسی به سیستم‌های خارجی را کاهش می‌دهد و سطح امنیت SQL Server را بالا می‌برد.

۲.۶: غیرفعال کردن Remote Access

گزینه Remote Access به کاربران امکان می‌دهد که از راه دور روی SQL Server عملیات‌های ذخیره‌شده[[21]](#footnote-21) اجرا کنند. غیرفعال کردن آن از حملات DoSجلوگیری می‌کند و در عین حال امنیت سرور را با کاهش امکان دسترسی از منابع خارجی افزایش می‌دهد.

۲.۷: غیرفعال کردن Remote Admin Connections

این گزینه به مدیران سرور این امکان را می‌دهد که از راه دور به سرور SQL متصل شوند، حتی اگر سرور در وضعیت غیرعادی باشد. غیرفعال کردن آن برای سرورهای غیرخوشه‌ای[[22]](#footnote-22) می‌تواند امنیت را افزایش دهد، هرچند در برخی موارد ممکن است نیاز به فعال بودن آن برای سرورهای خوشه‌ای باشد.

۲.۸: غیرفعال کردن Scan for Startup Procs

این ویژگی باعث می‌شود که SQL Server به‌طور خودکار رویه‌هایی را که برای اجرا در هنگام شروع سرویس تنظیم شده‌اند، اجرا کند. غیرفعال کردن آن می‌تواند از اجرای ناخواسته یا مخرب رویه‌ها جلوگیری کند و سطح حمله سرور را کاهش دهد.

۲.۹: غیرفعال کردن Trustworthy

ویژگی Trustworthy به پایگاه داده‌ها اجازه می‌دهد که به اشیاء پایگاه‌های داده دیگر دسترسی داشته باشند. غیرفعال کردن این ویژگی، باعث می‌شود که از سوءاستفاده‌های احتمالی CLR یا رویه‌های گسترش‌یافته جلوگیری شود و امنیت سرور افزایش یابد.

۲.۱۰: غیرفعال کردن پروتکل‌های غیرضروری SQL Server

SQL Server از پروتکل‌های مختلفی مانند TCP/IP، Named Pipes و Shared Memory برای ارتباطات استفاده می‌کند. غیرفعال کردن پروتکل‌های غیرضروری باعث کاهش سطح حمله به SQL Server می‌شود و از حملات از راه دور جلوگیری می‌کند.

۲.۱۱: استفاده از پورت‌های غیر استاندارد

پورت پیش‌فرض SQL Server برای ارتباطات TCP 1433 است. استفاده از پورت‌های غیر پیشفرض کمک می‌کند تا از حملات هدفمند روی پورت‌های شناخته‌شده جلوگیری شود و باعث می‌شود که شناسایی و دسترسی به سرور سخت‌تر شود.

۲.۱۲: غیرفعال کردن گزینه Hide Instance

این گزینه باعث می‌شود که نمونه[[23]](#footnote-23)‌های SQL Server در شبکه برای کاربران و ابزارهای شناسایی ظاهر نشوند. فعال کردن این گزینه امنیت را افزایش می‌دهد و دسترسی غیرمجاز به نمونه‌های SQL Server را کاهش می‌دهد.

۲.۱۳: غیرفعال کردن حساب کاربری sa

حساب 'sa' یک حساب پیش‌فرض با دسترسی کامل در SQL Server است که به راحتی هدف حملات brute force قرار می‌گیرد. غیرفعال کردن این حساب می‌تواند از دسترسی غیرمجاز به سرور جلوگیری کند و امنیت آن را به طور قابل توجهی افزایش دهد.

۲.۱۴: تغییر نام حساب کاربری sa

تغییر نام حساب 'sa' باعث می‌شود که این حساب شناخته‌شده کمتر در معرض حملات قرار گیرد. این تغییر نام، علاوه بر افزایش امنیت، امکان شناسایی و نفوذ به حساب را برای مهاجمان دشوارتر می‌کند.

۲.۱۵: غیرفعال کردن گزینه AUTO\_CLOSE

گزینه AUTO\_CLOSE باعث می‌شود که پایگاه داده پس از قطع ارتباط به‌طور خودکار بسته شود. این گزینه باید غیرفعال شود تا از بارگذاری مجدد غیرضروری پایگاه داده و مصرف اضافی منابع سرور جلوگیری شود.

۲.۱۶: اطمینان از عدم وجود حساب با نام sa

حتی اگر حساب 'sa' تغییر نام داده یا غیرفعال شده باشد، هیچ حساب دیگری نباید با این نام وجود داشته باشد. این اقدام از حملات Brute Force و دسترسی‌های غیرمجاز جلوگیری می‌کند و به حفظ امنیت سیستم کمک می‌کند.

۲.۱۷: تنظیم clr strict security به فعال

این گزینه باعث می‌شود که SQL Server برای اجرای اسمبلی‌های CLR از مجوزهای سختگیرانه‌تری استفاده کند و تنها اسمبلی‌های امن (SAFE) مجاز باشند. فعال کردن این ویژگی از حملات ناشی از کدهای CLR که می‌توانند آسیب‌زا باشند، جلوگیری می‌کند.

### 3-1-3-احراز هویت و مجوزها

این بخش به کنترل فرآیندهای احراز هویت[[24]](#footnote-24) و تخصیص مجوزها[[25]](#footnote-25) برای کاربران می‌پردازد.

۳.۱: تنظیم احراز هویت به Windows Authentication

استفاده از حالت Windows Authentication برای احراز هویت کاربران سرور SQL، که امنیت بیشتری نسبت به روش احراز هویت SQL Server ارائه می‌دهد.

۳.۲: لغو مجوز کاربر 'guest'

حذف مجوز اتصال برای کاربر 'guest' در تمام پایگاه‌های داده به‌جز پایگاه‌های اصلی (مانند master و tempdb) برای جلوگیری از دسترسی غیرمجاز.

۳.۳: حذف کاربران یتیم[[26]](#footnote-26)

کاربران پایگاه داده که دیگر به هیچ حساب کاربری مرتبط نیستند شناسایی و حذف می‌شوند تا از سوءاستفاده احتمالی جلوگیری شود.

۳.۴: عدم استفاده از احراز هویت SQL در پایگاه‌های مستقل

در پایگاه‌های داده مستقل[[27]](#footnote-27) به علت ملزم نکردن استفاده از پیچیدگی رمزعبور ،فقط از احراز هویت ویندوز استفاده شود و استفاده از احراز هویت SQL ممنوع گردد.

۳.۵: محدود کردن دسترسی حساب MSSQL Service

حساب MSSQL Service نباید دارای دسترسی ادمین باشد و دسترسی‌های آن باید فقط به موارد ضروری محدود شود.

۳.۶: محدود کردن دسترسی SQLAgent Service

حساب SQLAgent Service نیز نباید دارای مجوزهای ادمین باشد تا سطح دسترسی سرور کاهش یابد.

۳.۷: محدود کردن دسترسی Full-Text Service

دسترسی‌های حساب Full-Text Service باید به حداقل ضروری کاهش یابد تا امنیت افزایش یابد.

۳.۸: بررسی و حذف مجوزهای اضافی نقش عمومی

اطمینان حاصل شود که نقش عمومی فقط مجوزهای پیش‌فرض سیستم را دارد و هیچ مجوز اضافی به آن اختصاص داده نشده است.

۳.۹: حذف دسترسی گروه‌های BUILTIN ویندوز به SQL Server

گروه‌های BUILTIN ویندوز نباید به‌صورت پیش‌فرض به SQL Server دسترسی داشته باشند تا امنیت کلی افزایش یابد.

۳.۱۰: حذف دسترسی گروه‌های محلی ویندوز به SQL Server

اطمینان از اینکه هیچ گروه محلی ویندوز به SQL Server دسترسی ندارد مگر اینکه به‌طور خاص تنظیم شده باشد.

۳.۱۱: محدود کردن نقش عمومی در msdb

نقش عمومی نباید در پایگاه msdb به پراکسی‌های SQL Agent دسترسی داشته باشد تا از سوءاستفاده جلوگیری شود.

### 4-1-3- سیاست‌های رمز عبور

این تنظیمات برای افزایش امنیت حساب‌های کاربری توصیه می‌شود.

۴.۱: فعال کردن گزینه 'MUST\_CHANGE' برای تمامی ورودهای SQL Authenticated

این تنظیم تضمین می‌کند که کاربران هنگام ایجاد حساب جدید یا بازنشانی رمز عبور، رمز عبور خود را تغییر دهند.

۴.۲: فعال کردن گزینه 'CHECK\_EXPIRATION' برای ورودهای دارای نقش sysadmin

این گزینه، انقضای رمز عبور را برای ورودهای SQL فعال می‌کند و از استفاده طولانی‌مدت از رمزهای عبور جلوگیری می‌نماید.

۴.۳: فعال کردن گزینه 'CHECK\_POLICY' برای تمامی ورودهای SQL Authenticated

این تنظیم باعث می‌شود که قوانین امنیتی ویندوز (مانند پیچیدگی رمز عبور و حداقل طول) برای ورودهای SQL اعمال شود.

### 5-1-3- نظارت و ثبت گزارش‌ها

این بخش به ثبت و نظارت بر فعالیت‌های سیستم SQL Server می‌پردازد:

۵.۱: تنظیم تعداد فایل‌های گزارش خطا[[28]](#footnote-28) به حداقل ۱۲

تعداد فایل‌های گزارش خطا باید به ۱۲ یا بیشتر تنظیم شود تا بتوان تاریخچه بیشتری از رویدادها را برای بررسی مشکلات نگهداری کرد.

۵.۲: فعال کردن گزینه 'Default Trace'

این گزینه گزارش‌های پیش‌فرض SQL Server را فعال می‌کند و امکان بررسی تغییرات مهم در سرور، مانند ایجاد یا حذف اشیا، را فراهم می‌سازد.

۵.۳: تنظیم 'Login Auditing' به 'failed logins'

این تنظیم ورودهای ناموفق را ثبت می‌کند تا تلاش‌های غیرمجاز برای دسترسی شناسایی شوند.

۵.۴: استفاده از 'SQL Server Audit' برای ثبت ورودهای موفق و ناموفق

این ویژگی تمامی ورودهای موفق و ناموفق را ثبت می‌کند و اطلاعات جامعی از دسترسی‌ها ارائه می‌دهد.

### 6-1-3- توسعه نرم‌افزار

این بخش به توسعه و پیاده‌سازی نرم‌افزارهای کاربردی امن که با SQL Server ارتباط دارند، می‌پردازد:

۶.۱: اطمینان از پاکسازی ورودی‌های کاربر در پایگاه داده و برنامه‌های کاربردی

ورودی‌های کاربر باید به‌طور کامل بررسی و تصحیح شوند تا از حملات احتمالی مانند SQL Injection جلوگیری شود.

۶.۲: تنظیم سطح مجوز CLR Assembly Permission Set به SAFE\_ACCESS برای تمامی اسمبلی‌های CLR

برای افزایش امنیت، باید تمامی اسمبلی‌های CLR با سطح مجوز SAFE\_ACCESS اجرا شوند و از اجرای اسمبلی‌های خطرناک در سطح UNSAFE یا EXTERNAL\_ACCESS جلوگیری شود.

### 7-1-3- رمزنگاری

این بخش به استفاده از رمزنگاری برای حفاظت از داده‌ها در پایگاه داده و در طول انتقال می‌پردازد:

۷.۱: استفاده از الگوریتم رمزنگاری کلید متقارن AES\_128 یا بالاتر در پایگاه‌های داده غیرسیستمی

این تنظیم برای محافظت از داده‌ها با استفاده از الگوریتم‌های رمزنگاری قوی‌تر مانند AES\_128 توصیه می‌شود.

۷.۲: استفاده از اندازه کلید نامتقارن حداقل ۲۰۴۸ بیتی در پایگاه‌های داده غیرسیستمی

برای امنیت بیشتر، کلیدهای رمزنگاری نامتقارن باید حداقل ۲۰۴۸ بیت باشند.

۷.۳: رمزنگاری پشتیبان‌های پایگاه داده

پشتیبان‌ها باید رمزنگاری شوند تا در صورت دسترسی غیرمجاز به پشتیبان‌ها، داده‌ها محافظت شوند.

۷.۴: پیکربندی و فعال‌سازی رمزنگاری شبکه

شبکه باید برای انتقال امن داده‌ها رمزنگاری شود تا اطلاعات در حال انتقال ایمن باقی بمانند.

۷.۵: استفاده از رمزنگاری TDE برای پایگاه‌های داده

استفاده از TDE برای رمزنگاری داده‌های پایگاه‌داده در حین ذخیره‌سازی، برای محافظت در برابر دسترسی غیرمجاز به داده‌های ذخیره‌شده.

### 3-1-8- سایر ملاحظات

۸.۱: اطمینان از پیکربندی صحیح سرویس SQL Server Browser  
سرویس SQL Server Browser باید به‌درستی پیکربندی شود تا اطمینان حاصل شود که نمونه‌های SQL Server به درستی شناسایی می‌شوند و از دسترسی‌های غیرمجاز جلوگیری می‌شود.

فرایند سنجش مخاطرات‌

سنجش مخاطرات‌ به فرایند یافتن و گزارش کردن مخاطرات‌ گفته می‌شود. این ابزار به شما این امکان را می‌دهد که مشکلات امنیتی را تشخیص دهید و برطرف کنید.

یافتن مخاطرات‌ به دو صورت دستی و خودکار میتواند انجام شود و خروجی آن‌ یک گزارش است. در این گزارش گفته می‌شود که چه آسیب‌پذیری‌های در پایگاه داده وجود دارد. معمولا در این گزارش‌ها لیستی از آسیب‌پذیری‌ها وجود دارد که جزئیات هر آسیب‌پذیری، منبع آن، ریسک‌های آن و راه‌حل‌های رفع آن را نشان می‌دهد.

کاربرد‌های ابزار‌های سنجش مخاطرات‌

ابزارهای سنجش مخاطرات‌ امروزه یکی از ابزار‌های ضروری در زیرساخت‌های امنیتی سازمان‌ها می‌باشد. زمانی که یک مخاطره جدید یافت می‌شود، مدیر پایگاه داده می‌تواند یک ارزیابی انجام دهد، علت را بررسی کند و مراحل امن‌سازی را انجام دهد. زمانی که اصلاح و رفع مخاطره انجام شد، با سنجش مجدد می‌توان از رفع مخاطره موردنظر اطمینان حاصل کرد. این چرخه سنجش، امن‌سازی و باز‌ ارزیابی به فرایندی استاندارد برای سازمان‌ها تبدیل شده است تا اشکالات امنیتی خود را برطرف کنند.

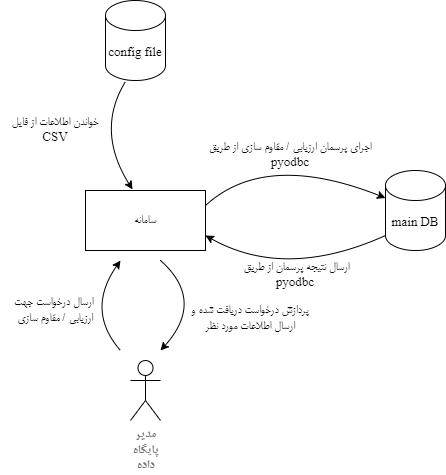
# معماری و پیاده‌سازی سامانه پیشنهادی

در این فصل به توضیح درباره معماری و پیاده‌سازی سامانه پیشنهادی می‌پردازیم. ابتدا معماری سامانه، بخش‌های اصلی آن و ارتباط میان این بخش‌ها را بررسی می‌کنیم. سپس نمودار کلاس هر بخش را آورده‌ایم. در ادامه، موارد کاربرد[[29]](#footnote-29) سامانه را تشریح می‌کنیم. در نهایت، پیاده‌سازی هر بخش سامانه را جداگانه بررسی می‌کنیم و کلاس‌های استفاده شده را توضیح می‌دهیم.

## 1-4-معماری سامانه پیشنهادی

این ابزار معماری ساده ای دارد، برنامه با واسط کاربری اطلاعات را از DBA میگیرد و با استفاده از رابط برنامه‌نویسی کاربردی (API) با پایگاه داده مورد نظر ارتباط برقرار می‌کند.  
اطلاعات مربوط به دستورات ارزیابی و مقاوم سازی از فایل config خوانده می‌شوند.

ارتباط میان اجزای مختلف سامانه را در شکل ‏4‑1 مشاهده می‌کنید.



شکل ‏4‑1: ارتباط میان اجزای مختلف سامانه

## 2-4-پیاده‌سازی

در این بخش، شیوه پیاده‌سازی ابزار بررسی می‌شود. پیاده‌سازی با زبان پایتون صورت گرفته است.

کتابخانه ها

لیست کتابخانه‌های مورد استفاده برای پیاده‌سازی عبارتند از :

* Pyodbc : به عنوان یک واسط بین برنامه و پایگاه داده عمل می‌کند.
* Tkinter : برای ساخت واسط کاربری برنامه استفاده شده است.
* tkhtmlView : برای خواندن و نمایش متن‌های html.
* CSV برای خواندن فایل csv.

### 1-2-4-پیکر‌بندی

برای اینکه این استاندارد را در قالب ورودی برنامه استفاده کنیم نیاز داشتیم تا یک طبقه بندی انجام دهیم تا ساختار فایل مناسب این کار شود، فایل ورودی اطلاعات در این برنامه config.csv نام دارد که ساختار آن مطابق با جدول 1 است.

|  |  |  |
| --- | --- | --- |
| **ردیف** | **نام ستون** | **توضیحات** |
| 0 | No. | شماره ردیف |
| 1 | Index | شماره امن‌سازی بر اساس ‌دسته‌بندی CIS |
| 2 | Dsc | توضیحات |
| 3 | AuditQuery | پرسمان ارزیابی |
| 4 | AuditType | نوع پرسمان ارزیابی (مطابق با اعداد جدول شماره 2) |
| 5 | ReturnNumbers | تعداد مقادیر برگشتی |
| 6 | Val1 | مقدار برگشتی 1 |
| 7 | Val2 | مقدار برگشتی 2 |
| 8 | Val3 | مقدار برگشتی 3 |
| 9 | RemType | نوع پرسمان امن‌سازی (مطابق با اعداد جدول شماره 3) |
| 10 | numberOfTransitions | تعداد ترنزاکسن ها |
| 11 | ReplacementName | نام متغییر جایگزین |
| 12 | Tsql1 | ترنز اکشن 1 |
| 13 | Tsql2 | ترنزاکشن 2 |
| 14 | Tsql3 | ترنز اکشن 3 |
| 15 | title | عنوان |

جدول 1 : ساختار فایل config

همچنین برای پیاده‌سازی پرسمان‌های ارزیابی آنها را مطابق جدول 2 طبقه بندی کردیم.

|  |  |
| --- | --- |
| کد ارزیابی | نحوه ارزیابی |
| 0 | این نوع باید بصورت دستی کنترل شود و پرسمان ندارد |
| 1 | یک ردیف با چند ستون که مقادیر آن باید 0 یا 1 باشند |
| 2 | هیچ مقداری نباید برگردانده شود |
| 3 | مقدار برگشتی باید از مقدار ثابتی بزرگتر باشد |
| 4 | مقدار برگشتی شامل چند ردیف است که مقادیر خاصی باید در آن وجود داشته باشد. |
| 5 | مقدار برگشتی نباید یک مقدار ثابت باشد |
| 6 | یک یا چند ردیف که نباید مقدار یکی از فیلدهایشان یک مقدار ثابت یاشد. |
| 7 | یک یا چند ردیف که باید مقدار یکی از فیلدهایشان یک مقدار ثابت یاشد. |

جدول 2 : طبقه‌بندی پرسمان‌های ارزیابی

برای پیاده‌سازی پرسمان‌های امن‌سازی ، طبقه‌بندی مطابق جدول 3 انجام شد.

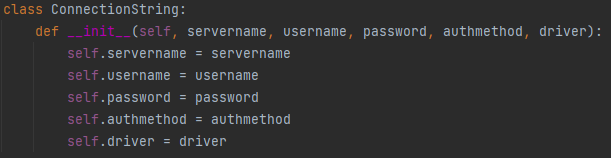
|  |  |
| --- | --- |
| کد امن‌سازی | نحوه امن‌سازی |
| 0 | این نوع باید بصورت دستی کنترل شود و پرسمان ندارد. |
| 1 | پرسمان از نوع ساده. |
| 2 | کاربر باید با توجه به مقادیری که در پرسمان ارزیابی برگردانده می‌شود، مقداری را در پرسمان مقاوم سازی واردکند. (مانند username ، login\_name) |

جدول 3 : طبقه‌بندی پرسمان‌های امن‌سازی

### 2-2-4-توصیف کلاس‌ها

در این بخش به توصیف کلاس‌های استفاده شده در پیاده‌سازی می‌پردازیم.

کلاس ConnectionString :  
در شکل 4-2مشخصات این کلاس مشاهده می‌شود، هر نمونه از این کلاس در واقعا اطلاعات مربوط به نام پایگاه داده و نام کاربری و روش ورود را نگه داری می‌کند.



شکل4- 2 : ConnectionString

کلاس Login :

مشخصات این کلاس را در شکل 4-3 مشاهده می‌شود. این کلاس برای ساخت فرم ورود استفاده می‌شود.



شکل 4-3: کلاس Login

توابع:   
loginControlFrame(): این تابع فرم ورود را می‌سازد.

detect\_driver() درایور‌های sql را شناسایی می‌کند

UserPassEnable() : غیرفعال کردن ورود username و password در حالتی که روش ورود Windows Authetication باشد.

Connect() ساخت connectionstring

کلاس AdminControls:

مشخصات این کلاس در شکل 4-3 مشاهده می‌شود، این کلاس برای نگهداری پارامترهای بدنه اصلی فرم، شامل دکمه ها، ورودی ها و جدول نتایج استفاده می‌شود.



شکل 3-4:کلاس AdminControls

توابع:  
adminControlsFrame() : ساخت فرم مربوط به دکمه‌ها و ورودی‌ها  
tableOutputFrame() : ساخت جدول موارد امن‌سازی

logout(): برای قطع ارتباط با پایگاه داده، خروج از صفحه اصلی و ورود به صفحه لاگین استفاده می‌شود. این تابع با کلیک کردن بر روی دکمه *Logout* فراخوانی می‌شود.

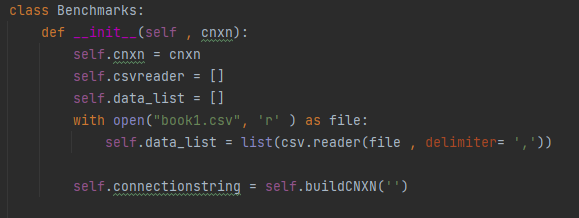
viewResults(): لیست پیشنهادات و نتایج آنها را نمایش/ بروز رسانی می‌کند. این تابع با کلیک کردن بر روی دکمه *View List* فراخوانی می‌شود.

getData(): اطلاعات مربوط به هر یک از موارد امن‌سازی را در قسمت *Description* نمایش میدهد ، این تابع با کلیک کردن بر روی هر ردیف از موارد امن‌سازی (بخش *List of Remediations*) فراخوانی می‌شود.

runRemediation() : اجرای امن‌سازی و نمایش نتایج آن در *remediation result .* این تابع با کلیک کردن بر روی دکمه *run remediation* فراخوانی می‌شود.

کلاس Bemchmarks:

مشخصات این کلاس در شکل 4-5 مشاهده می‌شود، این کلاس و توابع آن به طور کلی برای ارتباط با پایگاه داده ، خواندن فایل و اجرای پرسمان است.



شکل 4-5: کلاس Benchmarks

توابع:

buildCNXN() : اتصال به پایگاه داده با تنظیمات داده شده و از طریق pyodbc .

get\_databases() : دریافت لیست پایگاه داده ها جهت نمایش به کاربر.

handle\_sql\_variant\_as\_string(): خروجی را به فرمت مناسب تبدیل می‌کند.

runAudit(): برای اجرای پرسمان مربوط به ارزیابی‌ها

runRemediation() : برای اجرای پرسمان مربوط به مقاوم سازی‎ها.

## 3-4-ارزیابی نتایج

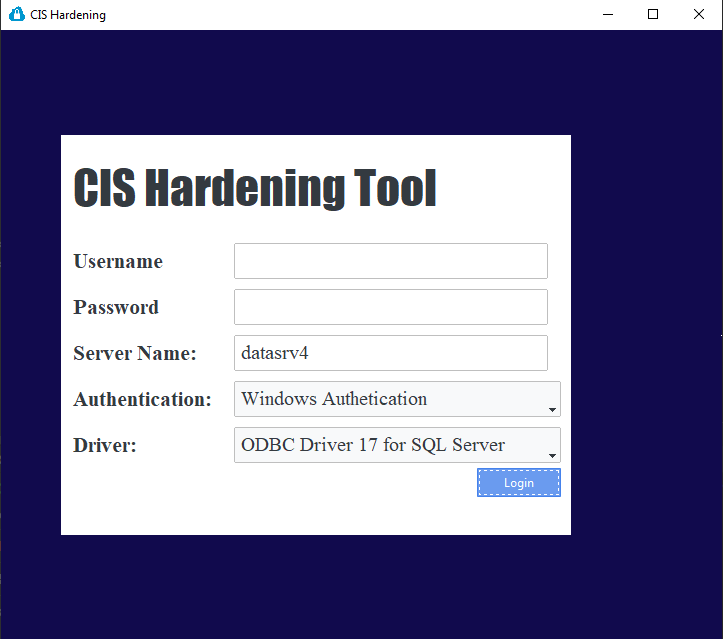
همانطور که قبل‌تر گفته شد، نحوه بررسی برخی از پیشنهادات ارزیابی به صورت دستی و استنتاجی میباشد (رجوع شود به جدول 2 : طبقه بندی پرسمان‌های ارزیابی) و در نتیجه مورد آزمون قرار نمیگیرد. اما این پیشنهادات به صورت نمایشی در سامانه قرار گرفته است و میتوان با مطالعه توضیحات مربوط به هر کدام ، اقدام به ارزیابی و امن‌سازی نمود.

در مجموع 45 پیشنهاد ارزیابی تهیه شده که از این تعداد 9 مورد دستی و بقیه خودکار هستند. در ادامه به نحوه کار با سامانه میپردازیم.

در ابتدا از نصب بودن یک نسخه odbc driver بر روی سیستم عامل اطمینان حاصل می‌کنیم.

برای اجرا برنامه نیاز به فایل config دارد که باید در کنار برنامه قرار گیرد.

پس از اجرای برنامه، با صغحه لاگین روبرو میشویم که مدیر پایگاه داده باید مشخصات اتصال به پایگاه داده را وارد کند، نحوه ورود هم به صورت windows authentication و هم sql server authentication انجام می‌شود است. در صورتی که درایور مورد نیاز تشخیص داده نشود از ورود به سامانه جلوگیری می‌شود.



شکل 4-4: فرم ورود

پس از ورود به برنامه با صفحه‌ای شامل از قسمت‌های مختلف روبرو می‌شویم که در ادامه به توصیف هر یک از این قسمت‌ها می‌پردازیم:

Database : یک لیست آبشاری از پایگاه داده‌ها را نمایش می‌دهدکه از آن پایگاه داده مورد نظر را انتخاب می‌کنیم.  
List of recommendations: لیستی از پیشنهادات امن‌سازی که شامل شماره، عنوان و نتیجه‎ی ارزیابی میباشد.  
Description : در این قسمت، توضیحات مربوط به هر پیشنهاد نمایش داده می‌شود که شامل عنوان، سطح مخاطره، توصیفی از مخاطره ، دلایل به وجود آمدن خطر، نحوه کشف مخاطره در پایگاه داده، نحوه امن‌سازی پایگاه داده در برابر این مخاطره، مقدار پیش‌فرض و منابع جهت مطالعه بیشتر است.  
Remediation result : جهت نمایش خروجی پرسمان ها استفاده می‌شود.  
View list : نمایش پیشنهادات و به روز کردن نتیجه ارزیابی.  
Run remediation : اجرای امن‌سازی برای هر پیشنهاد.  
Log out : قطع اتصال به پایگاه داده.

اکنون به مراحل کار با سامانه میپردازیم:

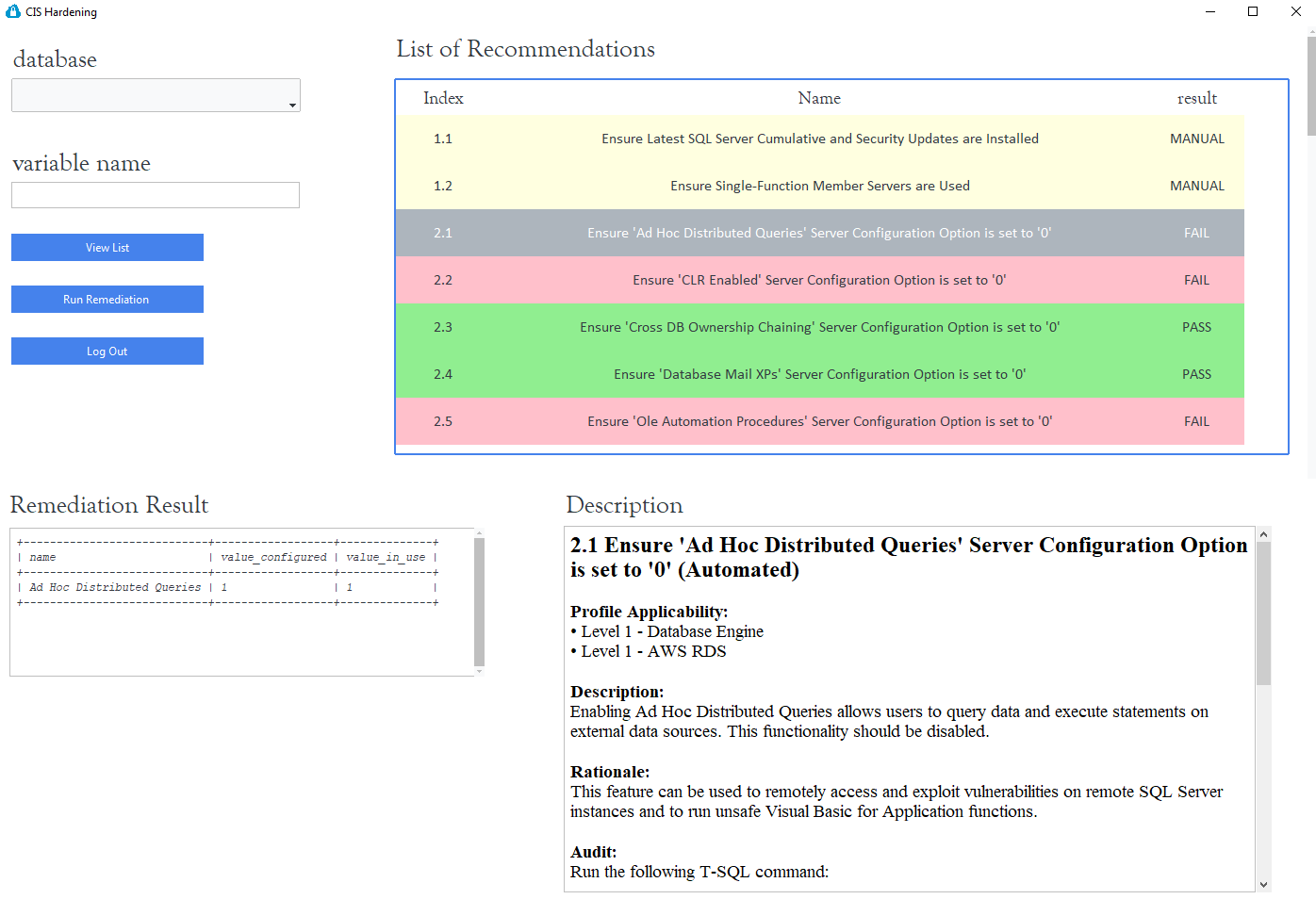
بخش ارزیابی:

پس از ورود موفق به برنامه، با انتخاب پایگاه داده و کلیک بر روی دکمه view list ، ارزیابی انجام می‌شود و نتیجه نمایش داده می‌شود.

در شکل4- 7 لیست پیشنهادات را ملاحظه میکنید، در صورتی که در تست ارزیابی قبول شده باشد به رنگ سبز و در صورتی که رد شده باشد به رنگ قرمز در میآید. در قسمت Description توضیحات مربوط به هر پیشنهاد نمایش داده شده است، این توضیحات همچنین در قسمت Remediation result نتیجه‌ی پرسمان ارزیابی نمایش داده می‌شود.

بخش امن‌سازی:

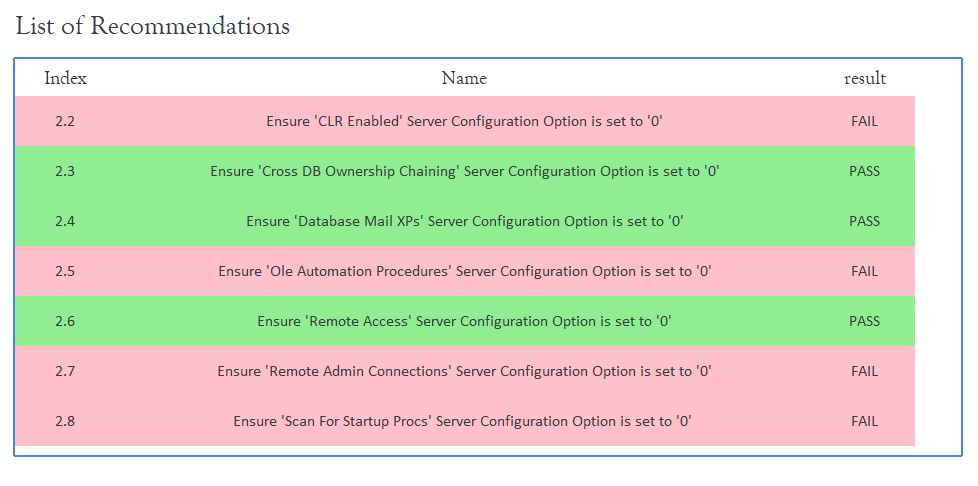
در صورتی که بتوان امن‌سازی را به صورت خودکار انجام داد دکمه‌ی run remediation فعال می‌شود و با کلیک روی این دکمه امن‌سازی انجام می‌شود توجه کنید که در برخی موارد برای امن‌سازی لازم است تا خروجی پرسمان ارزیابی را در قسمت variable name وارد شود (امن‌سازی نوع 2 در جدول 3 ). پس از آن مجددا نتیجه پرسمان در بخش Remediation result قابل مشاهده است.



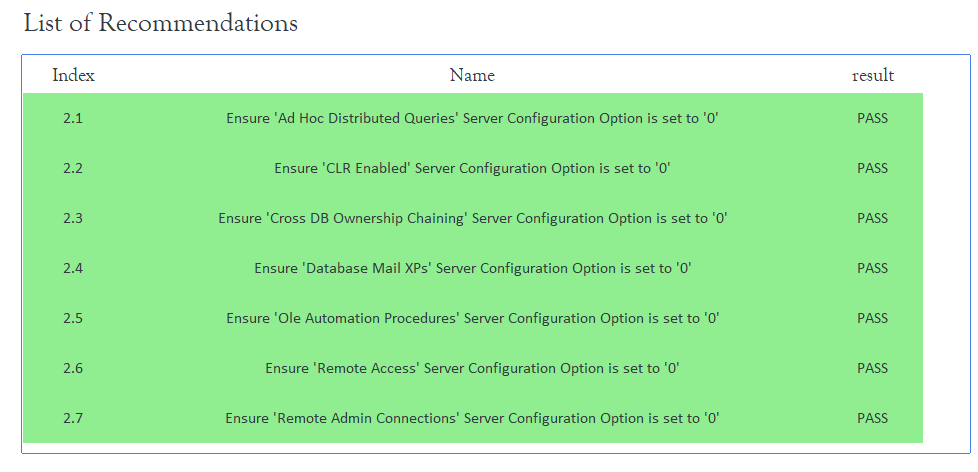
شکل 4-5: نمای کلی برنامه

همچنین با زدن دکمه log out اتصال به پایگاه داده قطع می‌شود و کاربر به صفحه ورود ارجاع داده می‌شود.

در شکل 4-8 و شکل 4-9 قبل و بعد از اجرای امن‌سازی روی یک پایگاه داده مشاهده می‌شود.



شکل 4-6: وضعیت ارزیابی امنیتی قبل از اجرای امن‌سازی



شکل 4-7: وضعیت ارزیابی بعد از اجرای امن‌سازی

# جمع بندی و پیشنهادات

در این پژوهش، موضوع امن‌سازی پایگاه داده SQL Server بر اساس استانداردهای CIS بررسی شد. به طور خاص اهمیت امنیت پایگاه داده و نقش SQL Server در مدیریت داده‌های حساس سازمانی بیان شد. تهدیدات امنیتی SQL Server بررسی شد و استانداردهای امنیتی برای مقابله با این تهدیدات معرفی شدند. پیشنهادات امن‌سازی استاندارد CIS برای SQL Server بطور مختصر توصیف شد. معماری و روش پیاده‌سازی سامانه پیشنهادی را توضیح دادیم. کار با سامانه و نتایج بدست امده را شرح دادیم.

نتایج پیاده‌سازی نشان داد که با اعمال این استانداردها، تعداد تهدیدات بالقوه و نقاط ضعف امنیتی به‌طور قابل توجهی کاهش می‌یابد.

**محدودیت‌ها:**

هر پروژه عملی با محدودیت‌هایی همراه است. در این پروژه، برخی از مهم‌ترین محدودیت‌ها عبارت بودند از:

محدودیت در محیط آزمایشی: پیاده‌سازی استاندارد CIS در یک محیط محدود انجام شد و ممکن است چالش‌هایی در محیط‌های بزرگ‌تر یا عملیاتی وجود داشته باشد.

عدم پوشش تمام تهدیدات امنیتی: تمرکز اصلی این پروژه بر روی تهدیدات مشخص و استاندارد CIS بود و امکان بررسی سایر روش‌های امنیتی وجود نداشت.

تأثیرگذاری بر عملکرد سیستم: برخی از تکنیک‌های امنیتی ممکن است تأثیر اندکی بر عملکرد SQL Server داشته باشند که در این پژوهش بررسی نشد.

**پیشنهادات برای کارهای آتی:**

این پروژه به برخی جنبه‌های امنیت SQL Server پرداخته است، اما زمینه‌های دیگری نیز وجود دارد که می‌تواند در پروژه‌های عملی آتی در این زمینه بررسی شود:

بررسی سایر استانداردها: مطالعه و مقایسه استانداردهای امنیتی دیگر مانند NIST یا ISO 27001 با CIS برای امن‌سازی SQL Server.

پیاده‌سازی در محیط‌های عملیاتی بزرگ‌تر: انجام آزمایش‌ها در محیط‌های واقعی با پایگاه‌های داده بزرگ‌تر و پیچیده‌تر برای بررسی اثربخشی استانداردهای CIS.

ارزیابی تأثیر بر عملکرد سیستم: بررسی تأثیر تکنیک‌های امنیتی بر سرعت و کارایی SQL Server و ارائه راهکارهایی برای بهینه‌سازی آن.

توسعه ابزارهای خودکار: پیشنهاد طراحی ابزارهایی که بتوانند به صورت خودکار تنظیمات استانداردهای CIS را در SQL Server پیاده‌سازی کنند.

**مراجع و مآخذ**

1.Internet Security Glossary, Version 2, RFC 4949 (Informational). R, Shirey. s.l. : IETF, 2007.

2.Security Econometrics The Dynamics of (In)Security. S, Frei. Zurich, Switzerland : ETH Zurich, 2009.

3. CIS Benchmarks List, <https://www.cisecurity.org/cis-benchmarks>, Accessed Sep 2024

4. Microsoft SQL documentation, https://learn.microsoft.com/en-us/sql/ ,Accessed Nov 2024

5.Database Security and Encryption: A Survey Study, Iqra Basharat, International Journal of Computer, May 2014

6. Database Security—Concepts, Approaches, and Challenges, Elisa Bertino, IEEE Transactions on Dependable and Secure Computing 2(1):2- 19 , Feb 2005



Department of Computer Engineering

B.Sc. Thesis

**Design and Implementation of SQL Server Hardening Tool**

**By:**

Reyhane Dehghan

**Supervisor:**

Dr Morteza Amini

Dec 2024

1. injection [↑](#footnote-ref-1)
2. Denial of service (DOS) [↑](#footnote-ref-2)
3. Distributed Denial of Service (DDOS) [↑](#footnote-ref-3)
4. Access Control [↑](#footnote-ref-4)
5. Principle of Least Privilege [↑](#footnote-ref-5)
6. Multi Factor Authorization (MFA) [↑](#footnote-ref-6)
7. Encryption [↑](#footnote-ref-7)
8. Man in the Middle [↑](#footnote-ref-8)
9. Transparent Data Encryption (TDE) [↑](#footnote-ref-9)
10. Backup [↑](#footnote-ref-10)
11. Recovery [↑](#footnote-ref-11)
12. Audinting [↑](#footnote-ref-12)
13. Logging [↑](#footnote-ref-13)
14. National Institute of Standards and Technology [↑](#footnote-ref-14)
15. Security Technical Implementation Guide [↑](#footnote-ref-15)
16. Center of Internet Security [↑](#footnote-ref-16)
17. Controls [↑](#footnote-ref-17)
18. Benchmarks [↑](#footnote-ref-18)
19. Patch [↑](#footnote-ref-19)
20. Common Language Runtime [↑](#footnote-ref-20)
21. Store Procedure [↑](#footnote-ref-21)
22. non-clustered [↑](#footnote-ref-22)
23. Instance [↑](#footnote-ref-23)
24. Authentication [↑](#footnote-ref-24)
25. Authorization [↑](#footnote-ref-25)
26. Orphan [↑](#footnote-ref-26)
27. Contained Databases [↑](#footnote-ref-27)
28. Error Log [↑](#footnote-ref-28)
29. Use Case [↑](#footnote-ref-29)