بسمه تعالی

سازمان فناوری اطلاعات ایران

معاونت امنیت فضای تولید و تبادل اطلاعات

مرکز ماهر

مقاوم‌سازی امنیتی MSSQL

پیشگفتار

در این گزارش، مقاوم‌سازی امنیتی SQL Server نسخه ۲۰۱۷ مورد بررسی قرار می‌گیرد. در این راستا، برای هر یک از پارامترهای امنیتی تاثیرگذار، شرح مختصری ارایه می‌گردد و سپس تهدید/توجیه امنیتی آن مورد بررسی قرار می‌گیرد. در نهایت نیز نحوه اطلاع از وضعیت فعلی پارامتر و چگونگی مقاوم‌سازی آن تشریح می‌گردد. بررسی پارامترهای مربوط به مقاوم‌سازی SQL Server در شش فصل متمایز صورت می‌گیرد. در فصل اول، نیازمندی‌های مربوط با امن‌سازی محیط اجرا ارائه می‌شود. فصل دوم به تشریح پارامترهای نصب و پیکربندی MSSQL می‌پردازد. فصل سوم به معرفی محدودیت‌هایی اختصاص دارد که می‌بایست بر روی فرآیند اتصال و ورود کاربران به MSSQL اعمال گردد. پارامترهای کنترل دسترسی و مجاز شماری در فصل چهارم مورد بررسی قرار می‌گیرند. پارامترهای مربوط به رویدادنگاری امن نیز در فصل پنجم بررسی می‌شوند. در فصل ششم، تنظیمات مربوط به رمزنگاری مطالعه می‌گردد و در پایان نیز، نحوه اجرای ابزار ارایه شده برای مقاوم سازی پایگاه‌داده تشریح می‌شود. لازم به ذکر است که در طول این مستند به تفاوت‌های میان سرور SQL اجرا شده بر روی ویندوز و سیستم عامل لینوکس و همچنین مفاهیم و دستورات متفاوت مربوط به هر یک از آنها اشاره شده است.

فهرست مطالب

۱ امن‌سازی محیط اجرا

در هر سیستم عامل، ابزارها و روش‌های مختلفی برای کنترل دسترسی وجود دارد. پیکربندی نرم‌افزارهایی چون پایگاه داده نیز باید بسیار دقیق و با رویکرد امنیتی انجام شود. با این حال، برای ایجاد یک مدل جامع برای محافظت از اطلاعات و سرویس‌های اطلاعاتی، باید از سطح سیستم‌عامل تمهیدات اندیشیده شود. چرا که اگر تنظیمات کارپذیر پایگاه داده به خوبی صورت گرفته باشد اما کنترل دسترسی بر روی فایل‌های آن اعمال نشده باشد، یک مهاجم می‌تواند به سادگی از داده‌ها نسخه‌برداری کرده و از این طریق اطلاعات زیادی فاش شود. در این راستا، در این فصل پیکربندی حساب‌های کاربری، فایل تنظیمات، دایرکتوری ذخیره داده و فایل‌های رویدادنگاری مورد بررسی قرار می‌گیرد.

۱-۱ پیکربندی فایل تنظیمات

تنظیمات اصلی MSSQL در فایل mssql.conf قرار دارد، از این رو حفاظت از این فایل مهم است.

تهدید/توجیه امنیتی:

از آنجایی که در این فایل تنظیمات اصلی MSSQL وجود دارد، تنها مالک این فایل یعنی root که مدیر سیستم است، حق تغییر این فایل را دارد. پس باید دقت کرد که اولاً مالک این فایل root باشد، ثانیا مجوز نوشتن تنها به مالک داده‌شده باشد.

اطلاع از وضعیت فعلی:

پس از نصب MSSQL بر روی سیستم عامل اوبونتو، مسیر پیش‌فرض برای فایل تنظیمات اصلی MSSQL، مسیر زیر است:

```
/var/opt/mssql/mssql.conf
```

با استفاده از دستور زیر می‌توان حقوق دسترسی مربوط به فایل تنظیمات اصلی MSSQL را مشاهده نمود:

```
ls -lh /var/opt/mssql/mssql.conf
```

خروجی این دستور (به صورت پیش‌فرض) به صورت زیر می‌باشد:

```
-rw-rw-r-- 1 mssql mssql 76 ۱۳:۴۳ ۵ ژوئن /var/opt/mssql/mssql.conf
```

مقاوم‌سازی:

دستورات زیر به ترتیب مالکیت، گروه کاربری مالکیت و حقوق دسترسی به فایل را به صورت امن مشخص می‌نمایند:

```
sudo  su
chown -R root /var/opt/mssql/mssql.conf
chgrp -R root /var/opt/mssql/mssql.conf
chmod 644 /var/opt/mssql/mssql.conf
```

۲ – ۱ پیکربندی دایرکتوری ذخیره داده

در پایگاه داده MSSQL، می‌توان مسیر پیش فرض داده‌های اصلی (مانند جداول و غیره) را با استفاده از دستور زیر به دست آورد:

```
SELECT SERVERPROPERTY('InstanceDefaultDataPath') AS
InstanceDefaultDataPath;
```

خروجی دستور فوق برای نسخه انتخابی گزارش به صورت زیر می‌باشد:

```
/var/opt/mssql/data/
```

تهدید/توجیه امنیتی:

برخلاف فایل‌های دیگر، مدیر پایگاه داده نباید مالک این فایل با شد. مالکیت فایل باید متعلق به کاربری بدون هرگونه حقوق ممتاز (مثلا کاربری با نام mssql) باشد؛ زیرا این کاربر اجازه انجام هیچ عملیاتی داخل سیستم لینوکس را ندارد. همچنین علاوه بر مدیر، هیچ‌کس دیگری نیز حق خواندن، تغییر یا اجرای این فایل را نباید داشته باشد. درنتیجه تمامی حقوق روی این فایل را از همه گرفته و مجوزهای مربوطه را تنها به کاربر mssql می‌دهیم.

اطلاع از وضعیت فعلی:

با استفاده از دستور زیر می‌توان حقوق دسترسی مربوط به دایرکتوری ذخیره داده را مشاهده نمود:

```
ls -lh /var/opt/mssql/data/
```

حقوق دسترسی، مالک و گروه کاربری مالکیت فایل های مربوط به پایگاه‌های داده پیش فرض موجود در دایرکتوری ذخیره داده به صورت زیر می‌باشند:

```
rw-r-----  mssql mssql
```

هنگامی که پایگاه‌داده جدیدی ایجاد می شود، فایل های مربوطه حقوق دسترسی متفاوتی نسبت به پایگاه‌های داده پیش فرض دارند:

```
rw-rw---- mssql mssql
```

انتظار می رود مجوزهای تمامی فایل‌ها داده (فایل‌هایی با پســـوند mdf) موجود در دایرکتوری ذخیره داده به صورت rw-r----- mssql mssql باشند.

مقاوم‌سازی:

د ستورات زیر به ترتیب مالکیت، گروه کاربری مالکیت و حقوق د ستر سی به داده‌های ا صلی (مانند جدوال و غیره) را به صورت امن تعیین می‌نماید.

```
sudo  su
chown -R mssql /var/opt/mssql/data
chgrp -R mssql /var/opt/mssql/data
chmod -R 640 /var/opt/mssql/data/
chmod 755 /var/opt/mssql/data/
```

۳ – ۱ پیکربندی فایل‌های رویدادنگاری

در MSSQL تمام رویدادهای مرتبط با پایگاه داده در فایل‌های از قبل تعیین شده‌ای ثبت می شوند. در پایگاه داده MSSQL، می‌توان مسیر پیش فرض فایل های رویدادنگاری را با استفاده از دستور زیر به دست آورد:

```
SELECT SERVERPROPERTY('InstanceDefaultLogPath') AS
InstanceDefaultLogPath;
```

خروجی دستور فوق برای نسخه انتخابی گزارش به صورت زیر می‌باشد:

```
/var/opt/mssql/data/
```

تهدید/توجیه امنیتی:

هیچ کاربری به جز mssql نباید حق خواندن یا نوشتن روی فایل‌های رویدادنگاری را داشته باشد. رعایت این مورد امنیتی از نشت اطلاعات این فایل‌ها جلوگیری می‌کند.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی مجوزهای مربوط به فایل رویدادنگاری، از دستور زیر استفاده می‌شود:

```
ls -lh /var/opt/mssql/data/
```

حقوق دســـترســـی، مالک و گروه کاربری مالکیت فایل های مربوط به پایگاه‌های داده پیش فرض موجود در دایرکتوری فایل‌های رویدادنگاری به صورت زیر می‌باشند:

```
rw-r-----  mssql mssql
```

هنگامی که پایگاه‌داده جدیدی ایجاد می شـــود، فایل های مربوطه حقوق دســـترســـی متفاوتی نســـبت به پایگاه‌های داده پیش فرض دارند:

```
rw-rw---- mssql mssql
```

انتظار می رود مجوزهای تمامی فایل‌های رویدادنگاری (فایل‌هایی با پســـوند ldf) موجود در دایرکتوری فایل‌های رویدادنگاری به صورت rw-r----- mssql mssql باشند.

مقاوم‌سازی:

د ستورات زیر به ترتیب مالکیت، گروه کاربری مالکیت و حقوق د ستر سی به فایل را به صورت امن م شخص می‌نمایند.

```
sudo  su
chown -R mssql /var/opt/mssql/data
chgrp -R mssql /var/opt/mssql/data
chmod -R 640 /var/opt/mssql/data/
chmod 755 /var/opt/mssql/data/
```

۴ – ۱ غیرفعال کردن حساب کاربری sa

حساب کاربری sa، حساب کاربری شناخته شده‌ای است که اغلب از آن استفاده می شود و دارای مجوزهای sysadmin است. این حساب کاربری در طول نصب ایجاد می شود. برای این حساب کاربری، principal_id برابر یک و sid برابر 0x01 است.

تهدید/توجیه امنیتی:

با توجه به آنکه حساب کاربری sa شناخته شده است، مهاجم سعی می‌کند با انجام حمله brute force به رمز عبور این نام کاربری با مجوز sysadmin دسترسی پیدا کند.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر می‌توان از وضعیت فعلی کاربر sa مطلع گردید. در صورتی که دستور زیر سطری را به عنوان خروجی برگرداند، نشان‌دهنده آن است که حساب کاربری sa فعال است.

```
SELECT name, is_disabled
FROM sys.server_principals
```

```
WHERE sid = 0x01
AND is_disabled = 0;
```

مقاوم‌سازی:

با اجرای دستور زیر، می‌توان حساب کاربری sa را غیرفعال کرد.

```
USE [master]
GO
DECLARE @tsql nvarchar(max)
SET @tsql = 'ALTER LOGIN ' + SUSER_NAME(0x01) + ' DISABLE'
EXEC (@tsql)
GO
```

استفاده از حساب کاربری sa در برنامه‌های کاربردی و اسکریپت‌ها از نظر امنیتی درست نیست، بنابراین در صورتی که از این حساب کاربری در برنامه‌ها استفاده شده باشد، با غیرفعال کردن آن، برنامه کاربردی یا اسکریپت نمی‌تواند به سرور پایگاه‌داده اصالت خود را اثبات نماید و در نتیجه اجرای آن‌ها با مشکل روبرو می‌شود. بنابراین غیرفعال کردن آن باید با توجه به این نکته صورت پذیرد.

۵-۱ تغییر نام حساب کاربری sa

حساب کاربری sa، حساب کاربری شناخته شده‌ای است که اغلب از آن استفاده می شود و دارای مجوزهای sysadmin است. این حساب کاربری در طول نصب ایجاد می شود. برای این حساب کاربری، principal_id برابر یک و sid برابر 0x01 است.

تهدید/توجیه امنیتی:

در صورتی که نام کاربری sa تغییر داده شود و برای مهاجم شناخته شده نباشد، اجرای حملات حدس رمز عبور و brute force مشکل خواهد شد.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر می‌توان فهمید که آیا نام کاربری sa تغییر داده شده است یا خیر. در صورتی که پرسمان زیر نام sa را برگرداند، بدین معناست که حساب کاربری sa، تغییر نام نداشته است.

```
SELECT name
FROM sys.server_principals
WHERE sid = 0x01;
```

مقاوم‌سازی:

با اجرای دستور زیر می‌توان نام حساب کاربری sa را تغییر داد.

```
ALTER LOGIN sa WITH NAME = <different_user>;
```

۶ – ۱ حذف حساب کاربری sa

حساب کاربری sa، حساب کاربری شناخته شده‌ای است که اغلب از آن استفاده می شود. بنابراین حتی در صورتی که حساب کاربری اصلی sa (principal_id = 1) تغییر نام داده شده باشد، نباید حساب کاربری با نام sa تعریف شود.

تهدید/توجیه امنیتی:

با اطمینان از عدم وجود نام کاربری sa، مهاجم قادر نخواهد بود بر روی نام کاربری شناخته شده، حملات brute force را اجرا کند.

اطلاع از وضعیت فعلی:

با اجرای پرسمان زیر می‌توان متوجه شد که آیا حساب کاربری با نام sa وجود دارد یا خیر. در صورتی که پرسمان زیر هیچ سطری برنگرداند، بدین معناست که حساب کاربری با نام sa وجود ندارد.

```
SELECT principal_id, name,
FROM sys.server_principals
WHERE name = 'sa';
```

مقاوم‌سازی:

با توجه به مقدار principal_id مربوط به حساب کاربری، می‌توان دستور ALTER یا DROP را اجرا کرد. لازم به ذکر است که در صورتی که مقدار پارامتر principal_id برابر یک باشد یا حساب کاربری مذکور مالک اشیای پایگاه‌داده باشد، نام sa را باید تغییر داد. در صورتی که حساب کاربری مالک هیچ شیئی در پایگاه‌داده نباشد می‌توان آن را حذف کرد.

```
USE [master]
GO
-- If principal_id = 1 or the login owns database objects, rename the
sa login
ALTER LOGIN [sa] WITH NAME = <different_name>;
GO
-- If the login owns no database objects, then drop it
-- Do NOT drop the login if it is principal_id = 1
DROP LOGIN sa
```

۱-۷ جمع‌بندی

در این فصل به تشریح پارامترهای امنیتی محیط اجرای سمپاد که به طور مستقیم بر عملکرد آن تاثیرگذار هستند، پرداختیم. در این راستا، تنظیمات مربوط به حقوق دسترسی فایل تنظیمات، دایرکتوری ذخیره‌داده، پیکره‌بندی فایل رویدادنگاری و امن سازی حساب کاربری sa که حساب کاربری شناخته شده‌ای است که اغلب از آن استفاده می‌شود و دارای مجوزهای sysadmin است، مورد بحث و بررسی قرار گرفت. مدیر سامانه به منظور بررسی پارامترهای مقاوم‌سازی و تهیه گزارش در این زمینه می‌تواند از چک لیست زیر استفاده نماید.

	عنوان	تنظیم صحیح	
		بله	خیر
۱	**ایمن سازی محیط اجرا**		
۱-۱	پیکربندی فایل تنظیمات	☐	☐
۱-۲	پیکربندی دایرکتوری ذخیره داده	☐	☐
۱-۳	پیکربندی فایل‌های رویدادنگاری	☐	☐
۱-۴	غیرفعال کردن حساب کاربری sa	☐	☐
۱-۵	تغییر نام حساب کاربری sa	☐	☐
۱-۶	حذف حساب کاربری با نام sa	☐	☐

۲ پیکربندی امن پایگاه‌داده

پارامترهای متعددی برای پیکربندی یک نمونه[1] از پایگاه‌داده MSSQL وجود دارد. پیکربندی ناصحیح این پارامترها می‌تواند مشکلاتی همچون منع ارائه سرویس و سرقت اطلاعات را به همراه داشته باشد. از اینرو تنظیم پارامترها باید با دقت و به درستی صورت پذیرد. در این بخش، برخی از مهم‌ترین پارامترهای امنیتی مربوط به پیکربندی MSSQL مورد بحث و بررسی قرار می‌گیرد.

۲-۱ پارامتر Ad Hoc Distributed Queries

فعال‌سازی Ad Hoc Distributed Queries به کاربر امکان پرس و جو و اجرای عبارات بر روی منابع داده خارجی را می‌دهد. این قابلیت می‌بایست غیرفعال شود (مقدار این پارامتر برابر صفر گردد). مقدار پیش‌فرض این پارامتر برابر صفر است.

تهدید/توجیه امنیتی:

از این قابلیت می‌توان برای دسترسی از راه دور و سوء استفاده از آسیب‌پذیری‌های موجود بر روی نمونه‌های SQL Server استفاده کرد.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر استفاده می‌شود. مقدار هر دو ستون (value_configured و value_in_use) می‌بایست برابر صفر باشد.

```
SELECT name,
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'Ad Hoc Distributed Queries';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را غیرفعال کرد.

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
```

[1] Instance

```
EXECUTE sp_configure 'Ad Hoc Distributed Queries', 0;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

۲-۲ پارامتر CLR Enabled

پارامتر CLR Enabled مشخص می‌کند که آیا اسمبلی[۲]های کاربر می‌توانند توسط SQL Server اجرا شوند یا خیر. این قابلیت باید غیرفعال شود (مقدار این پارامتر برابر صفر شود). مقدار پیش‌فرض این پارامتر برابر صفر است.

تهدید/توجیه امنیتی:

فعال سازی استفاده از اسمبلی‌های CLR، خطر حمله به SQL Server را افزایش می‌دهد و SQL Server از سوی اسمبلی‌های مخرب و ناخواسته در معرض خطر قرار می‌گیرد.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر استفاده می‌شود. مقدار هر دو ستون (value_configured و value_in_use) می‌بایست برابر صفر باشد.

```
SELECT name,
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'clr enabled';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را غیرفعال کرد.

```
EXECUTE sp_configure 'clr enabled', 0;
```

[۲] اسمبلی‌ها در واقع فایل‌های DLL ای هستند که در نمونه پایگاه‌داده MSSQL به کار می‌روند و مواردی همچون توابع، رویه‌های ذخیره شده و انواع داده‌های تعریف شده توسط کاربر را فراهم می‌کنند.

```
RECONFIGURE;
```

۳-۲ پارامتر Cross DB Ownership Chaining

پارامتر Cross DB Ownership Chaining زنجیره مالکیت متقابل پایگاه‌داده را در میان تمامی پایگاه‌های داده در سطح نمونه یا سرور کنترل می‌کند. این قابلیت باید غیرفعال شود (مقدار این پارامتر برابر صفر شود). مقدار پیش‌فرض این پارامتر برابر صفر است.

تهدید/توجیه امنیتی:

در صورتی که این پارامتر فعال باشد، یک عضو از نقش db_owner در یک پایگاه‌داده اجازه خواهد داشت به اشیایی که متعلق به حساب کاربری در پایگاه‌داده دیگری است، دسترسی پیدا کند و بدین ترتیب افشای اطلاعات غیرضروری پیش می‌آید. در صورت لزوم، این قابلیت باید برای پایگاه‌های داده مشخصی فعال شود ولی فعال کردن آن در سطح نمونه برای تمامی پایگاه‌های داده از نظر امنیتی درست نیست. با استفاده از دستور زیر می‌توان این قابلیت را برای پایگاه‌های داده مورد نظر فعال کرد. لازم به ذکر است که مقدار این پارامتر را نمی‌توان برای پایگاه‌های داده سیستمی master، model یا tempdb تغییر داد.

```
ALTER DATABASE <database_name> SET DB_CHAINING ON
```

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از پرسمان زیر استفاده می‌گردد. مقدار هر دو ستون (value_configured و value_in_use) باید برابر صفر باشد.

```
SELECT name,
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'cross db ownership chaining';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را غیرفعال کرد.

```
EXECUTE sp_configure 'cross db ownership chaining', 0;
RECONFIGURE;
GO
```

۲-۴ پارامتر Database Mail XPs

پارامتر Database Mail XPs قابلیت تولید و ارسال پست‌های الکترونیکی از SQL Server را کنترل می‌کند. این قابلیت باید غیرفعال شود (مقدار این پارامتر برابر صفر شود). مقدار پیش‌فرض این پارامتر نیز برابر صفر است.

تهدید/توجیه امنیتی:

با غیرفعال‌سازی این پارامتر، سطح حمله به SQL Server کاهش یافته وحمله DOS و کانالی برای ارسال داده از سرور پایگاه‌داده به میزبان راه دور حذف می‌شود.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر بهره می‌گیریم. مقدار هر دو ستون (value_configured و value_in_use) حاصل از اجرای این دستور باید برابر صفر باشد.

```
SELECT name,
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'Database Mail XPs';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را غیرفعال کرد.

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'Database Mail XPs', 0;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

۵ – ۲ پارامتر Ole Automation Procedures

پارامتر Ole Automation Procedures کنترل می‌کند که آیا می‌توان از اشـــیای OLE Automation در دسته‌های Transact-SQL نمونه‌هایی ایجاد کرد یا خیر. اشیای OLE Automation در واقع رویه‌های ذخیره شـــده توســـعه یافته بوده که به کاربر اجازه اجرای توابع خارج از SQL Server را می‌دهند. این قابلیت باید غیرفعال شود (مقدار این پارامتر برابر صفر شود). مقدار پیش‌فرض این پارامتر نیز برابر صفر است.

تهدید/توجیه امنیتی:

با فعال‌ســـازی این پارامتر خطر حمله به SQL Server افزایش یافته و کاربران قادر خواهند بود که توابعی در رابطه با امنیت SQL Server را اجرا نمایند.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر استفاده می‌شود. مقدار هر دو ستون (value_configured و value_in_use) می‌بایست برابر صفر باشد.

```
SELECT name,
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'Ole Automation Procedures';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را غیرفعال کرد.

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'Ole Automation Procedures', 0;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

[3] Batch

۲-۶ پارامتر Remote Access

پارامتر Remote Access امکان اجرای رویه‌های ذخیره شده محلی بر روی سرور از راه دور را کنترل می‌کند. این قابلیت باید غیرفعال شود و مقدار این پارامتر برابر صفر باشد. مقدار پیش‌فرض این پارامتر برابر یک بوده و فعال است.

تهدید/توجیه امنیتی:

با اجرای حمله منع سرویس بر روی سروربا انتقال پردازش پرس و جو بر روی یک هدف، عملکرد می‌تواند با مشکل روبرو شود.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر بهره می‌گیریم. مقدار هر دو ستون (value_configured و value_in_use) حاصل از اجرای این دستور باید برابر صفر باشد.

```
SELECT name,
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'remote access';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را غیرفعال کرد. پس از اجرای دستورات زیر باید سرویس پایگاه‌داده[۴] مجدداً راه‌اندازی شود.

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'remote access', 0;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

[4] Database Engine

۲-۷ پارامتر Remote Admin Connections

پارامتر remote admin connections کنترل می‌کند که آیا یک برنامه کاربردی کلاینت بر روی یک کامپیوتر راه دور می‌تواند از قابلیت Dedicated Administrator Connection (DAC) استفاده کند یا خیر. این قابلیت باید غیرفعال شود و مقدار این پارامتر برابر صفر باشد. مقدار پیش‌فرض این پارامتر برابر صفر است.

تهدید/توجیه امنیتی:

قابلیت Dedicated Administrator Connection به مدیر سیستم این امکان را می‌دهد که به سرور در حال اجرا دسترسی داشته باشد و عبارات T-SQL را اجرا کرده یا سعی به عیب‌یابی و عیب‌زدایی مشکلات سرور نماید، حتی در شرایطی که سرور قفل است یا در وضعیت غیرعادی اجرا می شود و قادر به پاسخ‌گویی به اتصالات SQL Server Database Engine نیست. این قابلیت باید غیرفعال باشد مگر در صورتی که نمونه یک خوشه[۵] است.

اطلاع از وضعیت فعلی:

برای اطلاعات از وضعیت فعلی این پارامتر از دستور زیر استفاده می‌شود. در صورتی که دستور زیر، هیچ داده‌ای برنگرداند، نشان می‌دهد که نمونه یک خوشه است و نیازی نیست که مقدار این پارامتر برابر صفر باشد. در صورتی که داده‌ای برگردانده شود، مقدار هر دو ستون باید برابر صفر باشد.

```
USE master;
GO
SELECT name,
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'remote admin connections'
AND SERVERPROPERTY('IsClustered') = 0;
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را غیرفعال کرد.

```
EXECUTE sp_configure 'remote admin connections', 0;
RECONFIGURE;
```

[5] Cluster

```
GO
```

۸-۲ پارامتر Scan For Startup Procs

در صورتی که پارامتر scan for startup procs فعال باشد، تمامی رویه‌های ذخیره شده که باید هنگام راه‌اندازی سرویس اجرا شوند، توسط SQL Server اسکن و اجرا می‌شوند. این قابلیت باید غیرفعال شود و مقدار این پارامتر برابر صفر باشد. مقدار پیش‌فرض این پارامتر نیز برابر صفر است.

تهدید/توجیه امنیتی:

غیرفعال‌سازی این پارامتر می‌تواند اهداف مخرب و تهدیدات امنیتی را کاهش دهد.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر بهره می‌گیریم. مقدار هر دو ستون (value_configured و value_in_use) حاصل از اجرای این دستور باید برابر صفر باشد.

```
SELECT name,
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'scan for startup procs';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را غیرفعال کرد. پس از اجرای دستورات فوق باید سرویس پایگاه‌داده مجدداً راه‌اندازی شود.

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'scan for startup procs', 0;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

۲-۹ پارامتر Trustworthy

پارامتر پایگاه‌داده TRUSTWORTHY به اشیای پایگاه‌داده اجازه می‌دهد که به اشیای سایر پایگاه‌های داده تحت شرایط مشخص دسترسی داشته باشند. این قابلیت باید غیرفعال باشد. مقدار پیش‌فرض این پارامتر OFF است مگر برای پایگاه‌داده msdb که باید ON باشد.

تهدید/توجیه امنیتی:

غیرفعال‌سازی این پارامتر، حفاظت در برابر اسمبلی‌های CLR مخرب و رویه‌های توسعه یافته را فراهم می‌کند.

اطلاع از وضعیت فعلی:

با اجرای پرس و جوی زیر لیستی از پایگاه‌های داده‌ای که پارامتر trustworthy برای آن‌ها فعال است، نشان داده می‌شود. پرس و جوی زیر نباید هیچ سطری بازگرداند.

```
SELECT name
FROM sys.databases
WHERE is_trustworthy_on = 1
AND name != 'msdb';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را بر روی پایگاه داده مورد نظر غیرفعال کرد.

```
ALTER DATABASE [<database_name>] SET TRUSTWORTHY OFF;
```

۲-۱۰ پارامتر TCP port

به یک نمونه SQL Server پیش‌فرض، پورت پیش‌فرض TCP:1433 برای ارتباطات TPC/IP تخصیص داده می‌شود. پورت TCP:1433 پورت شناخته شده‌ای برای SQL Server است و باید تغییر داده شود.

تهدید/توجیه امنیتی:

با تغییر پورت به مقدار غیرپیش‌فرض، می‌توان از پایگاه‌داده در مقابل حملاتی که مستقیم به پورت پیش‌فرض وارد می‌شوند، حفاظت کرد.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر بهره می‌گیریم. دستور زیر نباید سطری برگرداند. با وجود آنکه دستور زیر مربوط به خواندن پیکربندی رجیستری ویندوز است ولی می توان از آن برای استخراج پورت کارگزار SQL در سیستم عامل لینوکس نیز استفاده کرد.

```
DECLARE @value nvarchar(256);
EXECUTE master.dbo.xp_instance_regread
N'HKEY_LOCAL_MACHINE',
N'SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQLServer\SuperSocketNetLib\Tcp\IPAll',
N'TcpPort',
@value OUTPUT,
N'no_output';
SELECT @value AS TCP_Port WHERE @value = '1433';
```

مقاوم‌سازی:

در پنجره کنسول SQL Server Configuration Manager، وارد قسمت SQL Server Network Configuration و سپس Protocols for <instance name> شده و بر روی TCP/IP کلیک شود. در پنجره TCP/IP Properties در تب IP Addresses به قسمت IPALL رفته و فیلد TCP Port را از ۱۴۳۳ به پورت غیراستانداردی تغییر دهید. حال در همان پنجره SQL Server Configuration Manager بر روی SQL Server Services کلیک کرده و بر روی SQL Server (<instance name>) راست کلیک شود و Restart را انتخاب کنید. بدین ترتیب سرور SQL متوقف شده و مجدداً راه‌اندازی می‌شود. برای تغییر پورت سرور SQL در سیستم عامل لینوکس می‌توان از دستور زیر استفاده کرد.

```
/opt/mssql/bin/mssql-conf set network.tcpport <new_port>
```

پس از اجرای دستور فوق، سرویس MSSQL با استفاده از دستور زیر مجدداً راه اندازی شده تا تغییر پورت اعمال شود.

```
systemctl restart mssql-server.service
```

۲-۱۱ پارامتر Hide Instance

نمونه‌های SQL Server غیرخوشه‌ای در محیط‌های عملیاتی باید مخفی باشند تا توسط سرویس SQL Server Browser یافت نشوند. به صورت پیش‌فرض نمونه‌های SQL Server مخفی نیستند.

سرویس SQL Server Browser در حال حاضر در سیستم عامل لینوکس پشتیبانی نمی شود و در نتیجه، مخفی کردن نام نمونه‌های کارگزار SQL معنا ندارد.

تهدید/توجیه امنیتی:

با مخفی کردن نمونه‌های SQL Server، در وضـــعیت امن‌تری قرار می‌گیرند. توجه به این نکته حائز اهمیت است که نمونه‌های خوشه‌ای را نمی‌توان مخفی کرد.

اطلاع از وضعیت فعلی:

برای اطلاع از و ضعیت فعلی این پارامتر، از مجموعه د ستورات زیر بهره می‌گیریم. خروجی د ستورات زیر باید برابر مقدار یک باشد.

```
DECLARE @getValue INT;
EXEC master..xp_instance_regread
@rootkey = N'HKEY_LOCAL_MACHINE',
@key = N'SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQLServer\SuperSocketNetLib',
@value_name = N'HideInstance',
@value = @getValue OUTPUT;
SELECT @getValue;
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان نمونه SQL Server را مخفی کرد.

```
EXEC master..xp_instance_regwrite
@rootkey = N'HKEY_LOCAL_MACHINE',
@key = N'SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQLServer\SuperSocketNetLib',
@value_name = N'HideInstance',
@type = N'REG_DWORD',
@value = 1;
```

۲ – ۱۲ پارامتر xp_cmdshell

پارامتر xp_cmdshell کنترل می‌کند که آیا رویه ذخیره شـــده xp_cmdshell می‌تواند توســـط کاربر احراز اصالت شده برای اجرای دستورات سیستم عامل استفاده شود یا خیر. این قابلیت باید غیرفعال شود و مقدار این پارامتر برابر صفر باشد. مقدار پیش‌فرض این پارامتر نیز برابر صفر است.

تهدید/توجیه امنیتی:

رویه‌ی xp_cmdshell معمولاً توسط مهاجمین برای خواندن یا نوشتن داده از/بر روی سیستم عاملی که سرور پایگاه‌داده بر روی آن قرار دارد، استفاده می‌شود.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر بهره می‌گیریم. مقدار هر دو ستون (value_configured و value_in_use) حاصل از اجرای این دستور باید برابر صفر باشد.

```
SELECT name,
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'xp_cmdshell';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را غیرفعال کرد.

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'xp_cmdshell', 0;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

۲-۱۳ پارامتر AUTO_CLOSE

پارامتر AUTO_CLOSE تعیین می‌کند که آیا یک پایگاه‌داده پس از اتمام یک اتصال بسته شود یا خیر. در صورتی که این پارامتر فعال باشد، برای برقراری اتصالات بعدی به پایگاه‌داده، ابتدا پایگاه‌داده باید مجدداً بازگشایی شود. این قابلیت باید غیرفعال باشد. مقدار پیش‌فرض این پارامتر OFF است.

تهدید/توجیه امنیتی:

با توجه به آنکه احراز اصالت کاربران برای پایگاه‌های داده contained[۶]، در سطح پایگاه‌داده و نه در سطح سرور/نمونه انجام می‌شود، پایگاه‌داده باید همیشه برای احراز اصالت کاربران باز باشد. باز و بسته کردن مکرر پایگاه‌داده منابع بیشتری از سرور را مصرف کرده و می‌تواند منجر به منع از سرویس شود.

اطلاع از وضعیت فعلی:

با استفاده از دستور زیر می‌توان پایگاه‌های داده contained که AUTO_CLOSE برای آن‌ها فعال است را یافت. پرسمان زیر نباید سطری را به عنوان خروجی برگرداند.

```
SELECT name, containment, containment_desc, is_auto_close_on
FROM sys.databases
WHERE containment <> 0 and is_auto_close_on = 1;
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان این پارامتر را برای پایگاه‌داده مورد نظر غیرفعال کرد.

```
ALTER DATABASE <database_name> SET AUTO_CLOSE OFF;
```

۲-۱۴ مجموعه مجوزهای اسمبلی CLR

در صورتی که مجموعه مجوزهای اسمبلی CLR با مقدار SAFE_ACCESS تنظیم شده باشد، اسمبلی‌ها قادر نخواهند بود به منابع خارجی سیستم همچون فایل‌ها، شبکه، متغیرهای محیطی و رجیستری دسترسی پیدا کنند.

تهدید/توجیه امنیتی:

اسمبلی‌هایی که مجموعه مجوزهای آن‌ها با مقدار EXTERNAL_ACCESS یا UNSAFE تنظیم شده باشند، می‌توانند به قسمت‌های حساس سیستم عامل دسترسی پیدا کنند، داده‌ها را سرقت کنند یا آن‌ها را ارسال نمایند و وضعیت سیستم عامل را تغییر دهند.

اطلاع از وضعیت فعلی:

تمام اسمبلی‌های خروجی پرسمان زیر، باید مقدار SAFE_ACCESS در ستون permission_set_desc داشته باشند. توجه به این نکته حائز اهمیت است که بر روی اسمبلی‌هایی که توسط مایکروسافت ایجاد

[۶] پایگاه‌داده contained پایگاه داده‌ای است که از سایر پایگاه‌های داده و از نمونه پایگاه‌داده MSSQL که پایگاه‌داده بر روی آن قرار دارد، مجزا است.

شده‌اند (is_user_defined = 0)، چنین بررسی‌هایی انجام نمی‌شود و باید بتوانند به تمام سیستم دسترسی داشته باشند.

```
SELECT name,
permission_set_desc
FROM sys.assemblies
where is_user_defined = 1;
```

مقاوم‌سازی:

با استفاده از پرسمان زیر می‌توان permission_set برای اسمبلی مورد نظر را به مقدار SAFE_ACCESS تغییر داد.

```
ALTER ASSEMBLY <assembly_name> WITH PERMISSION_SET = SAFE;
```

۲-۱۵ سرویس پک و اصلاحیه‌ها

بسته‌های تکمیلی SQL Server شامل به‌روزرسانی‌هایی برای رفع مشکلات امنیتی و عملیاتی موجود در نرم‌افزار هستند. به یک وصله[7] تکی، اصلاحیه[8] گفته می‌شود. یک گروه کوچک از وصله‌ها، به‌روزرسانی تجمعی[9] و یک مجموعه بزرگ از وصله‌ها را سرویس پک[10] می‌نامند. نسخه‌ی SQL Server و بسته‌های تکمیلی باید آخرین موارد منطبق با نیازمندی‌های عملیاتی سازمان باشند.

تهدید/توجیه امنیتی:

نصب آخرین بسته‌های تکمیلی، احتمال سوء استفاده از آسیب پذیری‌های نرم‌افزار را کاهش می‌دهد.

اطلاع از وضعیت فعلی:

با استفاده از دستور زیر می‌توان سطح سرویس پک SQL Server را یافت. در خروجی حاصل از اجرای این دستور، ستون اول بیانگر سطح سرویس پک نصب شده و ستون دوم شماره ساخت[11] را نشان می‌دهد.

```
SELECT SERVERPROPERTY('ProductLevel') as SP_installed,
```

---

7 Patch
8 Hotfix
9 Cumulative Update
10 Service Pack
11 Build Number

```
SERVERPROPERTY('ProductVersion') as Version;
```

مقاوم‌سازی:

با ورود به لینک‌های زیر می‌توان آخرین بســته‌های تکمیلی مربوط به پایگاه‌داده MSSQL را دانلود و نصــب کرد.

```
--Download the appropriate patch and apply it to the installed version.
--The most recent SQL Server patches can be found here:
    • Hotfixes and Cumulative updates:
      http://blogs.msdn.com/b/sqlreleaseservices/
    • Service Packs: https://support.microsoft.com/en-us/kb/3177534
```

۲-۱۶ جمع‌بندی

در این فصــل به تشــریح برخی از مهم‌ترین پارامترهای امنیتی مربوط به پیکربندی ســمپاد MSSQL قبل از بکارگیری عملیاتی آن پرداختیم. در این راســتا برخی پارامترهای پیکربندی ســرور و پایگاه‌داده و بســته‌های تکمیلی SQL Server مورد بحث و بررسی قرار گرفت. مدیر سامانه به منظور بررسی پارامترهای مقاوم‌سازی و تهیه گزارش در این زمینه می‌تواند از چک لیست زیر استفاده نماید.

	عنوان	تنظیم صحیح	
		بله	خیر
۲	**پیکربندی امن پایگاه‌داده**		
۲-۱	پارامتر Ad Hoc Distributed Queries	☐	☐
۲-۲	پارامتر CLR Enabled	☐	☐
۲-۳	پارامتر Cross DB Ownership Chaining	☐	☐
۲-۴	پارامتر Database Mail XPs	☐	☐
۲-۵	پارامتر Ole Automation Procedures	☐	☐
۲-۶	پارامتر Remote Access	☐	☐
۲-۷	پارامتر Remote Admin Connections	☐	☐
۲-۸	پارامتر Scan For Startup Procs	☐	☐
۲-۹	پارامتر Trustworthy	☐	☐

۲-۱۰	پارامتر TCP port	☐	☐
۲-۱۱	پارامتر Hide Instance	☐	☐
۲-۱۲	پارامتر xp_cmdshell	☐	☐
۲-۱۳	پارامتر AUTO_CLOSE	☐	☐
۲-۱۴	مجموعه مجوزهای اسمبلی CLR	☐	☐
۲-۱۵	سرویس پک و اصلاحیه‌ها	☐	☐

۳ امن‌سازی اتصال به پایگاه‌داده

در این فصل به تنظیمات مربوط به احراز اصالت و خط‌مشی‌های مربوط به رمز عبور پرداخته می‌شود.

۳-۱ پارامتر Server Authentication

برای احراز اصالت کاربران بهتر است از حالت Windows Authentication استفاده شود. مقدار پیش‌فرض برای احراز اصالت کاربران در صورتی که MSSQL بر روی سیستم عامل ویندوز نصب شده باشد، حالت Windows Authentication است.

تهدید/توجیه امنیتی:

ویندوز مکانیزم احراز اصالت قوی‌تری را نسبت به SQL Server فراهم می‌کند.

اطلاع از وضعیت فعلی:

در صورتی که خروجی دستور زیر یک باشد، نشان‌دهنده آن است که برای ویژگی Server Authentication حالت Windows Authentication انتخاب شده است. مقدار صفر در خروجی، نشان‌دهنده احراز اصالت ترکیبی یعنی Windows Authentication و SQL Server Authentication است.

```
SELECT SERVERPROPERTY('IsIntegratedSecurityOnly') as [login_mode];
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان حالت احراز اصالت را به Windows Authentication تغییر داد. پس از اجرای دستور، به منظور اعمال تغییرات می‌بایست سرویس SQL Server مجدداً راه‌اندازی شود.

```
USE [master]
GO
EXEC xp_instance_regwrite N'HKEY_LOCAL_MACHINE',
```

```
N'Software\Microsoft\MSSQLServer\MSSQLServer',          N'LoginMode',
REG_DWORD, 1
GO
```

برای استفاده از حالت Windows authentication در صورت نصب SQL Server بر روی لینوکس، می‌توان از دستورالعمل موجود در لینک زیر استفاده کرد.

```
https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-active-
directory-authentication?view=sql-server-2017
```

۳-۲ حذف کاربران یتیم

کاربری که نمی‌تواند به نمونه SQL Server وارد شود، کاربر یتیم[1]نامیده شده و باید حذف شود.

تهدید/توجیه امنیتی:

کاربران یتیم می‌بایست حذف شوند تا راهی برای اهرم قرار دادن آنها برای سوء استفاده فراهم نباشد.

اطلاع از وضعیت فعلی:

به منظور شناسایی کاربران یتیم می‌بایست دستور زیر را بر روی هر یک از پایگاه‌های داده اجرا کرد.

```
USE [<database_name>];
GO
EXEC sp_change_users_login @Action='Report';
```

مقاوم‌سازی:

در صورتی که ورود کاربر یتیم به سامانه نیاز نباشد، می‌توان آن را از روی پایگاه‌داده حذف کرد.

```
USE [<database_name>];
GO
DROP USER <username>;
```

۳-۳ احراز اصالت در پایگاه‌های داده contained

پایگاه‌های داده contained قوانین مربوط به پیچیدگی رمز عبور را برای کاربران احراز اصالت شده اعمال نمی‌کنند.

---

1 Orphaned User 2

تهدید/توجیه امنیتی:

عدم وجود خط‌مشی رمز عبور، احتمال ایجاد اعتبار[3]ضعیف در پایگاه‌داده contained را افزایش می‌دهد.

اطلاع از وضعیت فعلی:

با اجرای پرس‌مان زیر در هر پایگاه‌داده contained می‌توان کاربران پایگاه‌داده که با استفاده از SQL Authentication احراز اصالت می‌شوند را پیدا کرد.

```
SELECT name AS DBUser
FROM sys.database_principals
WHERE name NOT IN ('dbo','Information_Schema','sys','guest')
AND type IN ('U','S','G')
AND authentication_type = 2;
GO
```

مقاوم‌سازی:

ا ستفاده از کاربران احراز ا صالت شده تو سط ویندوز در پایگاه‌های داده contained پی شنهاد می شود. بدین منظور می‌توان کاربری بر اساس کاربر ویندوزی با استفاده از دستور زیر ایجاد کرد [۳].

```
CREATE USER [<domain_name>\<user_name>];
```

همچنین می‌توان بر اساس گروه‌های ویندوزی، کاربری تعریف کرد.

```
CREATE USER [<domain_name>\<group_name>];
```

کاربران ایجاد شده به هیچ یک از لاگین‌های موجود در پایگاه‌داده master مرتبط نیستند و می‌توان از آن‌ها تنها در پایگاه‌داده contained استفاده کرد.

۳-۴ لاگین با گروه‌های پیشفرض

تا ق بل از SQL Server 2008، گروه BUILTIN\Administrators به عنوان لاگین[4]در SQL Server با مجوزهای sysadmin در طول نصب به صورت پیش‌فرض اضافه می‌شد.

---

[1] Credential [3]

[۱۴] لاگین‌ها در سطح سرور پایگاه‌داده ایجاد شده و به کاربران امکان اتصال به نمونه پایگاه‌داده MSSQL را می‌دهند.

تهدید/توجیه امنیتی:

گروه های پیش‌فرض همچون Administrators، Everyone، Authenticated Users و Guests عضـــو یت گســـترده‌ای دارند و نمی‌توان این اطمینان را حاصـــل کرد که تنها کاربران مورد نیاز به نمونه SQL Server دسترسی دارند. بنابراین، این گروه‌ها نباید هیچ سطح دسترسی به نمونه SQL Server داشته باشند.

اطلاع از وضعیت فعلی:

با استفاده از دستور زیر می‌توان متوجه شد که آیا گروه‌ها یا حساب‌های پیش‌فرض به عنوان لاگین‌های SQL Server اضافه شده‌اند یا خیر. پرسمان زیر نباید سطری را در خروجی برگرداند.

```
SELECT pr.[name], pe.[permission_name], pe.[state_desc]
FROM sys.server_principals pr
JOIN sys.server_permissions pe
ON pr.principal_id = pe.grantee_principal_id
WHERE pr.name like 'BUILTIN%';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان لاگین‌های مربوط به گروه‌های پیش‌فرض را از SQL Server حذف کرد. توجه به این نکته ضروری است که پیش از حذف لاگین‌های پیش‌فرض، بهتر است ابتدا گروهی در Active Directory شامل حساب‌های کاربری مورد نیاز ایجاد شود. در غیر این صورت ممکن است دسترسی به نمونه SQL Server کاملاً از بین برود.

```
USE [master];
GO
DROP LOGIN [BUILTIN\<name>];
GO
```

۵-۳ لاگین با گروه‌های محلی ویندوز

گروه‌های محلی ویندوز نباید به عنوان لاگین برای نمونه‌های SQL Server استفاده شوند.

تهدید/توجیه امنیتی:

در صورتی که گروه‌های محلی ویندوز به عنوان لاگین‌های SQL Server استفاده شوند، هر شخصی با مجوزهای مدیریتی در سطح سیستم عامل و بدون مجوز در سطح SQL Server می‌تواند کاربران را به گروه‌های محلی ویندوز اضافه کند و بدین ترتیب به آن‌ها دسترسی به نمونه SQL Server را بدهد.

اطلاع از وضعیت فعلی:

با استفاده از پرسمان زیر می‌توان متوجه شد که آیا گروه‌های محلی ویندوز به عنوان لاگین‌های سرور SQL اضافه شده‌اند یا خیر. دستور زیر نباید سطری را در خروجی برگرداند.

```
USE [master]
GO
SELECT pr.[name] AS LocalGroupName, pe.[permission_name], pe.[state_desc]
FROM sys.server_principals pr
JOIN sys.server_permissions pe
ON pr.[principal_id] = pe.[grantee_principal_id]
WHERE pr.[type_desc] = 'WINDOWS_GROUP'
AND pr.[name] like CAST(SERVERPROPERTY('MachineName') AS nvarchar) + '%';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان لاگین‌های مربوط به گروه‌های محلی ویندوز را از SQL Server حذف کرد. توجه به این نکته ضروری است که پیش از حذف لاگین‌های گروه‌های محلی بهتر است ابتدا گروهی در Active Directory شامل حساب‌های کاربری مورد نیاز ایجاد شود. در غیر این صورت، ممکن است دسترسی به نمونه SQL Server کاملاً از بین برود.

```
USE [master]
GO
DROP LOGIN [<name>]
GO
```

۳-۶ پارامتر MUST_CHANGE

در صورتی که گزینه MUST_CHANGE با مقدار ON تنظیم شده باشد، هنگامی‌که لاگین جدیدی ایجاد یا لاگین موجودی تغییر داده می‌شود، برای اولین ورود SQL Server درخواست به‌روزرسانی رمز عبور را می‌دهد.

تهدید/توجیه امنیتی:

با اجبار به تغییر رمز عبور پس از ایجاد لاگین جدید یا به‌روزرسانی لاگین موجود، از سوء استفاده از رمز عبور اولیه جلوگیری می‌شود.

اطلاع از وضعیت فعلی:

وارد SQL Server Management Studio شـده و به نمونه مورد نظر متصـل شـوید. در قسـمت Object Explore و در قسـمت Logins، بر روی لاگین مورد نظر راسـت کلیک کرده و گزینه Properties را انتخاب کنید. در پنجره باز شـده می توان بررسـی کرد که آیا گزینه User must change password at next login فعال ا ست یا خیر. توجه به این نکته ضروری ا ست که این گزینه باید بلافا صله پس از ایجاد یا تغییر لاگین برر سی و فعال شود. همچنین با ا ستفاده از د ستور زیر می توان گزینه MUST_CHANGE را برای لاگینی که اخیراً ایجاد شده است، بررسی کرد:

```
SELECT name, LOGINPROPERTY(name, 'IsMustChange') IsMustChange From
sys.sql_logins where name like '<login_name>';
```

مقاوم‌سازی:

برای ایجاد لاگین و فعال‌سازی MUST_CHANGE می‌توان از دستور زیر استفاده کرد.

```
CREATE LOGIN <login_name> WITH PASSWORD = '<password_value>'
MUST_CHANGE,
CHECK_EXPIRATION = ON, CHECK_POLICY = ON;
```

همان طور که در دسـتور فوق دیده می‌شـود، گزینه‌های CHECK_EXPIRATION و CHECK_POLICY با ید ON باشـ ند. همچنین هن گام به‌روزرسـانی یک لاگین و تغییر رمز عبور آن، می‌توان گزی نه MUST_CHANGE را فعال کرد.

```
ALTER LOGIN <login_name> WITH PASSWORD = '<new_password_value>'
MUST_CHANGE;
```

۳-۷ پارامتر CHECK_EXPIRATION

با انتخاب گزینه CHECK_EXPIRATION، همان خط‌مشـی انقضـای رمز عبوری که در ویندوز وجود دارد برای رمز عبور لاگین‌های موجود در SQL Server نیز اعمال می‌شود.

تهدید/توجیه امنیتی:

با اعمال خط‌مشی انقضای رمز عبور می‌توان مطمئن بود که رمز عبور مربوط به لاگین‌های SQL با مجوزهای sysadmin مرتباً تغییر می‌کنند و با استفاده از حمله brute force نمی‌توان به آن‌ها دسترسی پیدا کرد.

CONTROL SERVER یک مجوز معادل sysadmin است و لاگین‌هایی که دارای مجوز CONTROL SERVER نیز هستند باید خط‌مشی انقضای رمز عبور برای آن‌ها فعال باشد.

اطلاع از وضعیت فعلی:

با اســـتفاده از دســـتور زیر می‌توان لاگین‌هایی با مجوز sysadmin یا CONTROL SERVER که برای آن‌ها CHECK_EXPIRATION غیرفعال است را پیدا کرد.

```
SELECT l.[name], 'sysadmin membership' AS 'Access_Method'
FROM sys.sql_logins AS l
WHERE IS_SRVROLEMEMBER('sysadmin',name) = 1
AND l.is_expiration_checked <> 1
UNION ALL
SELECT l.[name], 'CONTROL SERVER' AS 'Access_Method'
FROM sys.sql_logins AS l
JOIN sys.server_permissions AS p
ON l.principal_id = p.grantee_principal_id
WHERE p.type = 'CL' AND p.state IN ('G', 'W')
AND l.is_expiration_checked <> 1;
```

مقاوم‌سازی:

برای تک تک لاگین‌هایی که خروجی پرسمان فوق هستند، با استفاده از دستور زیر می‌توان خط‌مشی انقضای رمز عبور را فعال کرد.

```
ALTER LOGIN [<login_name>] WITH CHECK_EXPIRATION = ON;
```

۳-۸ پارامتر CHECK_POLICY

با انتخاب گزینه CHECK_POLICY، همان خط‌مشـــی پیچیدگی رمز عبور که در ویندوز وجود دارد، برای رمز عبور لاگین‌های موجود در SQL Server نیز اعمال می‌شود.

تهدید/توجیه امنیتی:

با اعمال خط‌مشی پیچیدگی رمز عبور می‌توان مطمئن بود که رمز عبور لاگین‌ها با استفاده از حمله brute force به راحتی در معرض خطر و شناسایی نیستند.

اطلاع از وضعیت فعلی:

با استفاده از دستور زیر می‌توان لیست کاربرانی که POLICY_CHECK برای آن‌ها غیرفعال است را به دست آورد.

```
SELECT name, is_disabled
FROM sys.sql_logins
WHERE is_policy_checked = 0;
```

در دستور فوق در صورتی که مقدار ستون is_disabled برابر یک باشد، نشان‌دهنده آن است که لاگین غیرفعال و غیرقابل استفاده است. دستور فوق نباید سطری را به عنوان خروجی برگرداند.

مقاوم‌سازی:

برای تک تک لاگین هایی که خروجی پرس‌مان فوق هستند، با استفاده از دستور زیر می‌توان CHECK_POLICY را فعال کرد.

```
ALTER LOGIN [<login_name>] WITH CHECK_POLICY = ON;
```

۳-۹ جمع‌بندی

در این فصل به تشریح تنظیمات مربوط به امن سازی اتصال به سمپاد MSSQL پرداختیم. در این راستا، برخی از پارامترهای مرتبط و گزینه‌های مربوط به خط‌مشی رمز عبور مورد بحث و بررسی قرار گرفت. مدیر سامانه به منظور بررسی پارامترهای مقاوم‌سازی و تهیه گزارش در این زمینه می‌تواند از چک لیست زیر استفاده نماید.

	عنوان	تنظیم صحیح	
		بله	خیر
۳	**امن‌سازی اتصال به پایگاه‌داده**		
۳-۱	پارامتر Server Authentication	☐	☐
۳-۲	حذف کاربران یتیم	☐	☐
۳-۳	احراز اصالت در پایگاه‌های داده contained	☐	☐
۳-۴	لاگین با گروه‌های پیشفرض	☐	☐
۳-۵	لاگین با گروه‌های محلی ویندوز	☐	☐
۳-۶	پارامتر MUST_CHANGE	☐	☐
۳-۷	پارامتر CHECK_EXPIRATION	☐	☐
۳-۸	پارامتر CHECK_POLICY	☐	☐

۴ کنترل دسترسی و مجازشماری

با در نظر گرفتن اصـــل اعطای حداقل مجوزها به عنوان یک اصـــل کلی در حوزه امنیت، هر کاربر باید تنها مجوزهایی را دارا باشد که واقعاً برای اجرای مسئولیت‌هایش به آنها نیاز دارد. در این فصل، مجوزهایی که یک کاربر، نقش یا حساب سرویسی به آن‌ها نیازی ندارد، معرفی می شوند و نحوه لغو چنین مجوزهایی نیز بیان می‌گردد.

۴-۱ مجوز CONNECT

مجوز CONNECT مربوط به کاربر guest برای اتصـــال به تمامی پایگاه‌های داده SQL Server به غیر از پایگاه‌های داده master، msdb و tempdb باید حذف شود.

تهدید/توجیه امنیتی:

با سلب مجوز CONNECT از کاربر guest می‌توان مطمئن شد که یک لاگین نمی‌تواند به اطلاعات پایگاه‌داده دسترسی پیدا کند مگر آنکه صراحتاً این دسترسی به آن داده شده باشد.

اطلاع از وضعیت فعلی:

با اجرای پرســـمان زیر در هر پایگاه‌داده، می‌توان متوجه شـــد که آیا کاربر guest در هر پایگاه‌داده مجوز CONNECT را دارد یا خیر. پرسمان زیر نباید سطری را در خروجی برگرداند.

```
USE [<database_name>];
GO
SELECT DB_NAME() AS DatabaseName, 'guest' AS Database_User,
[permission_name], [state_desc]
FROM sys.database_permissions
WHERE [grantee_principal_id] = DATABASE_PRINCIPAL_ID('guest')
AND [state_desc] LIKE 'GRANT%'
AND [permission_name] = 'CONNECT'
AND DB_NAME() NOT IN ('master','tempdb','msdb');
```

مقاوم‌سازی:

با اجرای دستور زیر، مجوز CONNECT از کاربر guest در پایگاه‌داده مورد نظر سلب می‌شود.

```
USE [<database_name>];
GO
REVOKE CONNECT FROM guest;
```

۲-۴ تنظیم حساب خدماتی سرویس MSSQL

حساب خدماتی[۴] استفاده شده توسط سرویس MSSQL SERVER در نمونه پیش‌فرض[۶] یا سرویس MSSQL$<InstanceName> در نمونه نامیده شده[۷] باید به طور مستقیم یا غیرمستقیم (از طریق گروه‌ها) عضوی از گروه مدیران ویندوز باشد. همچنین با توجه به آنکه مجوزهای حساب LocalSystem (نام مستعار برای NT AUTHORITY\SYSTEM) بیشتر از مجوزهای مورد نیاز سرویس SQL Server است، بنابراین نباید برای سرویس MSSQL استفاده شود.

هنگامی که MSSQL بر روی سیستم عامل Ubuntu نصب می شود، در مسیر /lib/system/system سرویسی با نام mssql-server.service ایجاد می شود. کاربری که این سرویس را اجرا می کند، کاربری با نام mssql است. برای اطمینان از اینکه سرویس مورد نظر توسط کدام کاربر اجرا می شود می توان محتوای فایل mssql-server.service را مشاهده کرد.

همچنین باید این اطمینان حاصل شود که کاربر mssql تنها عضوی از گروه mssql است و به سایر گروه‌های موجود در Ubuntu تعلق ندارد. برای بررسی این موضوع می توان از دستور زیر استفاده کرد:

```
groups mssql
```

خروجی دستور فوق mssql است که به معنی آن است که کاربر mssql تنها عضو گروه mssql است و عضو هیچ گروه دیگری نیست.

تهدید/توجیه امنیتی:

با توجه به قانون کمترین مجوز، حساب سرویسی نباید مجوزهایی بیش از آنچه که برای کار خود نیاز دارد، داشته باشد. به هنگام نصب و راه‌اندازی برای سرویس‌های SQL Server مجوزهای مورد نیاز به حساب سرویسی تخصیص داده می‌شوند و مجوزهای بیشتری نیاز نیست.

اطلاع از وضعیت فعلی:

باید این اطمینان حاصل شود که حساب سرویسی (در حالت محلی یا حالتی که حساب‌ها به صورت مرکزی مدیریت می‌شوند) عضوی از گروه مدیران سیستمی ویندوز نباشد.

---

[1] Service Account
[1] Default Instance
[1] Named Instance
[5]
[6]
[7]

بر روی سیستم‌عامل Ubuntu باید این اطمینان حاصل شود که کاربر mssql تنها عضوی از گروه mssql است و به سایر گروه‌های موجود در Ubuntu تعلق ندارد. برای بررسی این موضوع از دستور زیر استفاده می‌شود:

```
groups mssql
```

انتظار می رود، خروجی دستور فوق mssql باشد که به معنی آن است که کاربر mssql تنها عضو گروه mssql است و عضو هیچ گروه دیگری نیست.

مقاوم‌سازی:

در صورتی که از حساب LocalSystem استفاده شده است، باید آن را به حسابی با مجوزهای کمتر تغییر داد. در صورتی که حساب سرویسی عضوی از گروه مدیران سیستم باشد، باید آن را از گروه مدیران حذف کرد. بر روی سیستم عامل Ubuntu در صورتی که کاربر mssql عضو گروه‌هایی با مجوزهای بیش از حد نیاز باشد، با استفاده از دستور زیر می‌توان کاربر mssql را از عضویت گروه خارج کرد.

```
sudo deluser mssql <group_name>
```

۳-۴ تنظیم حساب خدماتی سرویس SQLAgent

حساب خدماتی[۸] استفاده شده توسط سرویس SQLSERVERAGENT در نمونه پیش‌فرض[۹] یا سرویس SQLAGENT$<InstanceName> در نمونه نامیده شده؛[۰] باید به طور مستقیم یا غیرمستقیم (از طریق گروه‌ها) عضوی از گروه مدیران ویندوز باشد. همچنین با توجه به آنکه مجوزهای حساب LocalSystem (نام مستعار برای NT AUTHORITY\SYSTEM) بیشتر از مجوزهای مورد نیاز سرویس SQLAGENT است، بنابراین نباید برای سرویس SQLAGENT استفاده شود.

SQL Server Agent در محیط لینوکس بخشی از بسته mssql-server است که به صورت پیش‌فرض غیرفعال است. فعال سازی SQL Server Agent با استفاده از دستورات زیر انجام شده و سرویس و کاربر اجراکننده‌ی مجزایی برای آن وجود ندارد.

```
sudo /opt/mssql/bin/mssql-conf set sqlagent.enabled true
sudo systemctl restart mysql-server
```

---

1 Service Account
1 Default Instance
2 Named Instance
8
9
0

بنابراین با توجه به آنکه سرویس مجزایی برای SQL Agent در لینوکس وجود ندارد، حساب سرویس مجزایی نیز برای اجرای آن وجود ندارد و پس از فعال‌سازی SQL Server Agent، اجرای سرویس mssql-server و کاربر اجراکننده‌ی mssql برای SQL Agent نیز کاربرد دارد.

تهدید/توجیه امنیتی:

با توجه به قانون کمترین مجوز، حساب خدماتی نباید مجوزهایی بیش از آنچه که برای کار خود نیاز دارد، داشته باشد. برای سرویس‌های SQL Server، هنگام نصب و راه‌اندازی، مجوزهای مورد نیاز به حساب خدماتی تخصیص داده می‌شوند و مجوزهای بیشتری نیاز نیست.

اطلاع از وضعیت فعلی:

باید این اطمینان حاصل شود که حساب خدماتی (در حالت محلی یا حالتی که حساب‌ها به صورت مرکزی مدیریت می‌شوند) عضوی از گروه مدیران سیستمی ویندوز نباشد.

مقاوم‌سازی:

در صورتی که از حساب LocalSystem استفاده شده است، باید آن را به حسابی با مجوزهای کمتر تغییر داد. در صورتی که حساب سرویسی عضوی از گروه مدیران سیستم باشد، باید آن را از گروه مدیران حذف کرد. بر روی سیستم عامل Ubuntu در صورتی که کاربر mssql عضو گروه‌هایی با مجوزهای بیش از حد نیاز باشد، با استفاده از دستور زیر می‌توان کاربر mssql را از عضویت گروه خارج کرد.

```
sudo deluser mssql <group_name>
```

۴ – ۴ تنظیم حساب خدماتی سرویس Full-Text

حساب خدماتی[1] استفاده شده توسط سرویس MSSQLFDLauncher در نمونه پیش‌فرض[2] سرویس MSSQLFDLauncher$<InstanceName> در نمونه نامیده شده[3] باید به طور مستقیم یا غیرمستقیم (از طریق گروه‌ها) عضوی از گروه مدیران ویندوز باشد. همچنین با توجه به آنکه مجوزهای حساب LocalSystem (نام مستعار برای NT AUTHORITY\SYSTEM) بیشتر از مجوزهای مورد نیاز سرویس Full-Text است، بنابراین نباید برای سرویس Full-Text استفاده شود.

---

1 Service Account
2 Default Instance
3 Named Instance

در سیستم عامل لینوکس، full-text search را می توان با استفاده از دستور زیر نصب کرد:

```
sudo apt-get install -y mssql-server-fts
```

پس از نصب، به منظور فعال سازی full-text search باید سرویس mssql-server مجدداً راه اندازی شود. این بدان معنا است که در لینوکس برای full-text search سرویس و کاربر اجراکننده‌ی مجزایی وجود ندارد.

بنابراین با توجه به آنکه سرویس مجزایی برای full-text search در لینوکس وجود ندارد، حساب سرویس مجزایی نیز برای اجرای آن وجود ندارد و پس از نصب و فعال‌سازی full-text search، اجرای سرویس mssql-server آن را فعال و به مرحله اجرا می‌رساند.

تهدید/توجیه امنیتی:

با توجه به قانون کمترین مجوز، حساب خدماتی نباید مجوزهایی بیش از آنچه که برای کار خود نیاز دارد، داشته باشد. به هنگام نصب و راه‌اندازی برای سرویس‌های SQL Server مجوزهای مورد نیاز به حساب خدماتی تخصیص داده می‌شوند و مجوزهای بیشتری نیاز نیست.

اطلاع از وضعیت فعلی:

باید این اطمینان حاصل شود که حساب سرویسی (در حالت محلی یا حالتی که حساب‌ها به صورت مرکزی مدیریت می‌شوند) عضوی از گروه مدیران سیستمی ویندوز نباشد.

مقاوم‌سازی:

در صورتی که از حساب LocalSystem استفاده شده باشد می‌بایست آن را به حسابی با مجوزهای کمتر تغییر داد. در صورتی که حساب سرویسی عضوی از گروه مدیران سیستم باشد، باید آن را از گروه مدیران حذف کرد. بر روی سیستم عامل Ubuntu در صورتی که کاربر mssql عضو گروه‌هایی با مجوزهای بیش از حد نیاز باشد، با استفاده از دستور زیر می‌توان کاربر mssql را از عضویت گروه خارج کرد.

```
sudo deluser mssql <group_name>
```

۵ – ۴ تنظیم مجوزهای نقش public

Public یک نقش سروری ثابت شامل تمامی لاگین‌ها است. برخلاف سایر نقش‌های سروری ثابت، مجوزهای نقش public را می‌توان تغییر داد. با توجه به قانون کمترین مجوز، به نقش public نباید مجوزهایی بیشتر از مجوزهای پیش‌فرض مشخص شده توسط مایکروسافت اعطا شود.

تهدید/توجیه امنیتی:

هر یک از لاگین‌های موجود در SQL Server متعلق به نقش public هســـتند و نمی‌توانند از این نقش حذف شوند. بنابراین هر مجوزی که به این نقش اعطا می‌شود، به تمامی لاگین‌ها اعطا خواهد شد مگر آنکه صراحتاً از لاگین‌های مشخص یا نقش‌های تعریف شده توسط کاربر سلب شوند.

اطلاع از وضعیت فعلی:

با ا ستفاده از پر سمان زیر می‌توان متوجه شد که آیا مجوزهای بی شتری به نقش public اعطا شده ا ست یا خیر. دستور زیر نباید سطری برگرداند.

```
SELECT *
FROM master.sys.server_permissions
WHERE (grantee_principal_id = SUSER_SID(N'public') and state_desc LIKE
'GRANT%')
AND NOT (state_desc = 'GRANT' and [permission_name] = 'VIEW ANY
DATABASE'
and class_desc = 'SERVER')
AND NOT (state_desc = 'GRANT' and [permission_name] = 'CONNECT' and
class_desc = 'ENDPOINT' and major_id = 2)
AND NOT (state_desc = 'GRANT' and [permission_name] = 'CONNECT' and
class_desc = 'ENDPOINT' and major_id = 3)
AND NOT (state_desc = 'GRANT' and [permission_name] = 'CONNECT' and
class_desc = 'ENDPOINT' and major_id = 4)
AND NOT (state_desc = 'GRANT' and [permission_name] = 'CONNECT' and
class_desc = 'ENDPOINT' and major_id = 5);
```

مقاوم‌سازی:

در صورتی که دستور فوق سطری برگرداند، مجوزهای غیراصلی باید به نقش‌های سروری تعریف شده توسط کاربر که به آن دسترسی نیاز دارند، اعطا شوند و از نقش public سلب گردند.

```
USE [master]
GO
REVOKE <permission_name> FROM public;
GO
```

۴-۶ تنظیم دسترسی نقش public به پروکسی‌های SQL Agent

نقش پایگاه‌داده‌ای public شامل تمامی کاربران در پایگاه‌داده msdb است. پروکسی‌های[۲] SQL Agent یک زمینه امنیتی[۳] تعریف می‌کنند که در آن یک مرحله‌ی کاری[۴] می‌تواند اجرا شود.

تهدید/توجیه امنیتی:

در صورتی که مجوز دسترسی به پروکسی‌های SQL Agent به نقش پایگاه‌داده‌ای public اعطا شده باشد، تمامی کاربران قادر به استفاده از پروکسی خواهند بود که ممکن است مجوز زیادی برای آن‌ها باشد و قانون کمترین مجوز نقض گردد.

اطلاع از وضعیت فعلی:

با استفاده از پرسمان زیر می‌توان متوجه شد که آیا دسترسی به پروکسی‌ها به نقش public موجود در پایگاه‌داده msdb اعطا شده است یا خیر. پرسمان زیر نباید سطری برگرداند.

```
USE [msdb]
GO
SELECT sp.name AS proxyname
FROM dbo.sysproxylogin spl
JOIN sys.database_principals dp
ON dp.sid = spl.sid
JOIN sysproxies sp
ON sp.proxy_id = spl.proxy_id
WHERE principal_id = USER_ID('public');
GO
```

مقاوم‌سازی:

با استفاده از پرسمان زیر می‌توان دسترسی به پروکسی‌های موجود در لیست فوق را از نقش public سلب کرد. توجه به این نکته ضروری است که پیش از سلب مجوز دسترسی به پروکسی از نقش public باید این مجوز به نقش‌های پایگاه‌داده‌ای تعریف شده توسط کاربر یا سایر لاگین‌ها اعطا شود. در غیر این صورت مراحل کاری SQL Agent مرتبط با دسترسی لغو شده با شکست روبرو می‌شوند.

---

2 Proxy
2 Security Context
2 Job Step
4
5
6

```
USE [msdb]
GO
EXEC dbo.sp_revoke_login_from_proxy @name = N'public', @proxy_name = N'<proxyname>';
GO
```

۴-۷ جمع‌بندی

در این فصل مجوزهایی که ممکن است به کاربر، نقش یا حساب سرویسی اعطا شود در حالی که نیازی به اعطای آن‌ها نیست مورد بحث و بررسی قرار گرفت. مدیر سامانه به منظور بررسی پارامترهای مقاوم سازی و تهیه گزارش در این زمینه می‌تواند از چک لیست زیر استفاده نماید.

	عنوان	تنظیم صحیح	
		بله	خیر
۴	**کنترل دسترسی و مجازشماری**		
۴-۱	مجوز CONNECT	☐	☐
۴-۲	تنظیم حساب خدماتی سرویس MSSQL	☐	☐
۴-۳	تنظیم حساب خدماتی SQLAgent	☐	☐
۴-۴	تنظیم حساب خدماتی سرویس Full-Text	☐	☐
۴-۵	تنظیم مجوزهای نقش public	☐	☐
۴-۶	تنظیم دسترسی نقش public به پروکسی‌های SQL Agent	☐	☐

۵ تنظیمات رویدادنگاری

ثبت وقایع سیستم و بازبینی آن‌ها در صورت رخداد مشکلات فنی و امنیتی یکی از نیازمندی‌های اصلی در پایگاه‌های داده است. با توجه به اینکه انواع وقایعی که در سیستم رخ می‌دهند از درجه اهمیت متفاوتی برخوردار هستند و ثبت کلیه وقایع بدون توجه به ارزش هر یک می‌تواند منجر به کاهش کارایی سرور و به هدر رفتن فضای دیسک گردد، لازم است مجموعه‌ای از وقایع مهم شناسایی گردد و رویدادنگاری در مورد آنها همواره انجام شود. در مورد رویدادنگاری دیگر وقایع جانبی، بسته به توانمندی‌های پردازشی و ذخیره‌سازی سرور می‌توان تصمیم‌گیری کرد. در این بخش به تشریح برخی از مهم‌ترین پارامترهای امنیتی مربوط به پیکربندی تنظیمات رویدادنگاری می‌پردازیم.

۵ – ۱ تنظیم تعداد فایل‌های رویدادنگاری خطا

فایل‌های رویدادنگاری خطا[۲] پیش از آنکه مجدداً بر روی آن‌ها نوشته شود می‌بایست از آن‌ها نسخه پشتیبان تهیه شده و از آنها حفاظت شود.

تهدید/توجیه امنیتی:

رویدادنگاری خطا در SQL Server شامل اطلاعات مفیدی از رویدادهای سرور و تلاش برای ورود به سرور است.

اطلاع از وضعیت فعلی:

با استفاده از دستور زیر می‌توان تعداد فایل‌های رویدادنگاری خطا در سرور را پیدا کرد. مقدار بازگشتی باید بزرگتر یا مساوی ۱۲ باشد.

```
DECLARE @NumErrorLogs int;
EXEC master.sys.xp_instance_regread
N'HKEY_LOCAL_MACHINE',
N'Software\Microsoft\MSSQLServer\MSSQLServer',
N'NumErrorLogs',
@NumErrorLogs OUTPUT;
SELECT ISNULL(@NumErrorLogs, -1) AS [NumberOfLogFiles];
```

[2] Error Log Files [7]

با وجود آنکه دستور فوق مربوط به خواندن پیکربندی رجیستری ویندوز است ولی می توان از آن برای استخراج تعداد فایل‌های رویدادنگاری خطاها در سیستم عامل لینوکس نیز استفاده کرد. خروجی دستور فوق بر روی Ubuntu برابر ۱۲۸ است یعنی مقدار پیش فرض حداکثر تعداد فایل های رویدادنگاری در لینوکس برابر ۱۲۸ است.

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان حداکثر تعداد فایل‌های رویدادنگاری خطا در SQL Server را به مقدار دلخواهی بزرگتر یا مساوی ۱۲ تغییر داد. تعداد فایل های رویداد نگاری را در سرور SQL اجرا شده بر روی لینوکس نمی‌توان تغییر داد [۲].

```
EXEC master.sys.xp_instance_regwrite
N'HKEY_LOCAL_MACHINE',
N'Software\Microsoft\MSSQLServer\MSSQLServer',
N'NumErrorLogs',
REG_DWORD,
<NumberAbove12>;
```

۵-۲ پارامتر Default Trace Enabled

ردیابی[۲۸] پیش‌فرض، امکان رویدادنگاری از فعالیت‌های پایگاه‌داده شامل ایجاد حساب‌های کاربری، افزایش امتیازها[۲۹] و اجرای دستورات DBCC را فراهم می‌کند. مقدار این پارامتر به صورت پیش‌فرض برابر یک (یعنی فعال) است.

تهدید/توجیه امنیتی:

ردیابی پیش‌فرض، اطلاعات با ارزشی در مورد فعالیت‌های امنیتی بر روی سرور را فراهم می‌کند.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از پرسمان زیر استفاده می‌کنیم. لازم به ذکر است که هر دو ستون حاصل از خروجی پرسمان می‌بایست مقداری برابر یک داشته باشند.

```
SELECT name,
```

---

[2] Trace
[2] Privilege Elevation
[8]
[9]

```
CAST(value as int) as value_configured,
CAST(value_in_use as int) as value_in_use
FROM sys.configurations
WHERE name = 'default trace enabled';
```

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان پارامتر Default Trace Enabled را فعال کرد.

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'default trace enabled', 1;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

۵-۳ پارامتر Login Auditing

تنظیم Login Auditing، تلاش‌های ناموفق برای احراز اصالت لاگین‌های SQL Server را ثبت می‌کند.

تهدید/توجیه امنیتی:

با ثبت تلاش‌های ناموفق برای ورود به SQL Server می‌توان حملات حدس رمز عبور را شناسایی کرد.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی پارامتر از پرسمان زیر استفاده می‌کنیم.

```
EXEC xp_loginconfig 'audit level';
```

در صورتی که مقدار بازگشتی در ستون config_value برابر failed باشد، تنها تلاش‌های ناموفق ثبت می‌شوند. در صورتی که مقدار بازگشتی در این ستون برابر all باشد، کلیه تلاش‌ها اعم از موفق و ناموق ثبت می‌شوند. هر دو تنظیم معتبر و درست هستند ولی با توجه به اینکه رویدادها در SQL ServerErrorlog ثبت می‌شوند بهتر است در میان رویدادهای مربوط به تلاش‌های ناموفق، رویدادهای ورود موفقیت آمیز ثبت نشوند.

مقاوم‌سازی:

با استفاده از دستور زیر می‌توان سطح ممیزی را برای ثبت رویدادهای مربوط به تلاش‌های ناموفق تغییر داد. پس از اجرای دستور زیر، نمونه SQL Server می‌بایست مجدداً راه‌اندازی شود. تغییر تنظیم AuditLevel در SQL Server اجرا شده بر روی لینوکس تا این زمان امکان‌پذیر نیست.

```
EXEC xp_instance_regwrite N'HKEY_LOCAL_MACHINE',
N'Software\Microsoft\MSSQLServer\MSSQLServer',          N'AuditLevel',
REG_DWORD, 2
```

۵-۴ تنظیم SQL Server Audit

ممیزی در SQL Server قادر است تمامی تلاش‌های موفق و ناموفق برای ورود به سرور را در یکی از سه محل زیر ثبت کند:

- در قسمت application در ویندوز
- در قسمت security در ویندوز
- فایل سیستم

تهدید/توجیه امنیتی:

با ثبت تلاش‌های ناموفق برای ورود به پایگاه‌داده می‌توان حملات حدس رمز عبور را شناسایی کرد. همچنین ثبت تلاش‌های موفق برای ورود به پایگاه‌داده در تحلیل‌های جرم‌شناسی کاربرد خواهند داشت.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی پارامتر از پرسمان زیر استفاده می‌کنیم.

```
SELECT
S.name AS 'Audit Name'
, CASE S.is_state_enabled
WHEN 1 THEN 'Y'
WHEN 0 THEN 'N' END AS 'Audit Enabled'
, S.type_desc AS 'Write Location'
, SA.name AS 'Audit Specification Name'
, CASE SA.is_state_enabled
WHEN 1 THEN 'Y'
WHEN 0 THEN 'N' END AS 'Audit Specification Enabled'
, SAD.audit_action_name
, SAD.audited_result
FROM sys.server_audit_specification_details AS SAD
JOIN sys.server_audit_specifications AS SA
```

```
ON SAD.server_specification_id = SA.server_specification_id
JOIN sys.server_audits AS S
ON SA.audit_guid = S.audit_guid
WHERE SAD.audit_action_id IN ('CNAU', 'LGFL', 'LGSD');
```

خروجی پر سمان فوق باید شامل سه سطر با شد که در ستون audit_action_name موارد زیر قرار دا شته باشد:

- AUDIT_CHANGE_GROUP
- FAILED_LOGIN_GROUP
- SUCCESSFUL_LOGIN_GROUP

در هر سه سطر مقدار ستون Audit Enabled و Audit Specification Enabled باید Y (به معنی فعال) باشد. همچنین مقدار ستون audited_result باید SUCCESS AND FAILURE باشد.

مقاوم‌سازی:

با ا ستفاده از د ستور زیر می‌توان ممیزی برای ثبت رویدادهای مربوط به تلاش‌های موفق و ناموفق برای ورود لاگین‌ها را فعال کرد.

```
CREATE SERVER AUDIT TrackLogins
TO APPLICATION_LOG;
GO
CREATE SERVER AUDIT SPECIFICATION TrackAllLogins
FOR SERVER AUDIT TrackLogins
ADD (FAILED_LOGIN_GROUP),
ADD (SUCCESSFUL_LOGIN_GROUP),
ADD (AUDIT_CHANGE_GROUP)
WITH (STATE = ON);
GO
ALTER SERVER AUDIT TrackLogins
WITH (STATE = ON);
GO
```

۵-۵ جمع‌بندی

در این فصل به تشریح برخی از مهم‌ترین پارامترهای امنیتی مربوط به پیکربندی تنظیمات رویدادنگاری پرداختیم. مدیر سامانه به منظور بررسی پارامترهای مقاوم‌سازی و تهیه گزارش می‌تواند از چک لیست زیر استفاده نماید.

	عنوان	تنظیم صحیح	
		بله	خیر
۵	**تنظیمات رویدادنگاری**		
۵-۱	تنظیم تعداد فایل‌های رویدادنگاری خطا	☐	☐
۵-۲	پارامتر Default Trace Enabled	☐	☐
۵-۳	پارامتر Login Auditing	☐	☐
۵-۴	تنظیم SQL Server Audit	☐	☐

۶ تنظیمات رمزنگاری

استفاده از رمزنگاری یک راه حل برای عدم دسترسی آشکار به داده‌ها برای افراد غیر مجاز است. در این فصل تعدادی از پارامترهای مهم مربوط به تنظیمات رمزنگاری در پایگاه‌داده معرفی می‌شوند.

۶ – ۱ الگوریتم رمزنگاری کلید متقارن

گزینه‌هایی که برای الگوریتم AES در پایگاه‌داده وجود دارد یعنی AES_128، AES_192 و AES_256 باید به عنوان الگوریتم رمزنگاری کلید متقارن استفاده شوند.

تهدید/توجیه امنیتی:

الگوریتم‌های رمزنگاری DES، DESX، RC2، RC4 و RC4_128 الگوریتم‌های ضعیف و منسوخ شده‌ای هستند و نباید در پایگاه‌داده استفاده شوند. استفاده از الگوریتم‌های رمزنگاری ضعیف و منسوخ شده سیستم را در معرض خطر شکسته شدن کلید توسط مهاجم قرار می‌دهد.

اطلاع از وضعیت فعلی:

با استفاده از کد زیر می‌توان متوجه شد که در کدامیک از پایگاه‌های داده، الگوریتم رمزنگاری کلید متقارن مناسبی انتخاب نشده است. پرسمان زیر باید در هر یک از پایگاه‌های داده کاربر اجرا شود. پرسمان زیر نمی‌بایست هیچ سطری را به عنوان خروجی برگرداند.

```
USE [<database_name>]
GO
SELECT db_name() AS Database_Name, name AS Key_Name
FROM sys.symmetric_keys
WHERE algorithm_desc NOT IN ('AES_128','AES_192','AES_256')
AND db_id() > 4;
GO
```

مقاوم‌سازی:

با استفاده از پرسمان ALTER SYMMETRIC KEY *نمی‌توان* الگوریتم رمزنگاری کلید متقارن استفاده شده را تغییر داد. شاید بهتر با شد که کلید با الگوریتم رمزنگاری ضعیف حذف و کلید با الگوریتم رمزنگاری قوی‌تری ایجاد شود.

به عنوان نمونه، می‌توان با استفاده از دستور زیر کلید متقارن با الگوریتم رمزنگاری مورد نظر را ایجاد کرد:

```
CREATE SYMMETRIC KEY <key_name>
WITH ALGORITHM = <algorithm>
```

```
ENCRYPTION BY <encrypting_mechanism>;
```

همچنین با استفاده از دستور زیر می‌توان کلید موجود را حذف کرد:

```
DROP SYMMETRIC KEY <symmetric_key_name>
```

۲ – ۶ طول کلید نامتقارن

پیشنهاد مایکروسافت برای طول کلیدهای رمزنگاری نامتقارن، کلیدهایی با طول حداقل ۲۰۴۸ بیت می‌باشد.

تهدید/توجیه امنیتی:

امن‌ترین انتخاب موجود در پایگاه‌داده MSSQL اســـتفاده از الگوریتم رمزنگاری RSA_2048 برای کلیدهای نامتقارن ا ست. انتخاب کلید با طول بزرگتر باعث افت کارایی می شود ولی احتمال شک سته شدن سی ستم رمزنگاری توسط مهاجم را کاهش می‌دهد.

اطلاع از وضعیت فعلی:

با اســتفاده از پرســمان زیر می‌توان متوجه شــد که در کدامیک از پایگاه‌های داده کاربر، طول کلید رمزنگاری منا سبی انتخاب ن شده ا ست. پر سمان زیر باید برای هر یک از پایگاه‌های داده کاربر اجرا شود. پر سمان زیر نباید هیچ سطری را به عنوان خروجی برگرداند.

```
USE <database_name>;
GO
SELECT db_name() AS Database_Name, name AS Key_Name
FROM sys.asymmetric_keys
WHERE key_length < 2048
AND db_id() > 4;
GO
```

مقاوم‌سازی:

با استفاده از پرسمان ALTER ASYMMETRIC KEY نمی‌توان طول کلید الگوریتم رمزنگاری نامتقارن استفاده شده را تغییر داد. شاید بهتر باشد که کلید با الگوریتم نامناسب حذف و کلیدی با الگوریتم و طول مناسب ایجاد شود. به عنوان نمونه می‌توان با استفاده از دستور زیر کلید نامتقارن با الگوریتم دلخواه را ایجاد کرد.

```
CREATE ASYMMETRIC KEY <key_name>
    WITH ALGORITHM = <algorithm>
    ENCRYPTION BY <encrypting_mechanism>;
```

```
<algorithm> ::=
      { RSA_4096 | RSA_3072 | RSA_2048 | RSA_1024 | RSA_512 }
```

همچنین با استفاده از دستور زیر می‌توان کلید موجود را حذف کرد.

```
DROP ASYMMETRIC KEY <key_name>;
```

۶-۳ جمع‌بندی

در این فصـــل به تشـــریح برخی از مهم‌ترین پارامترهای امنیتی مربوط به پیکربندی تنظیمات رمزنگاری پرداختیم. مدیر سامانه به منظور بررسی پارامترهای مقاوم‌سازی و تهیه گزارش در این زمینه می‌تواند از چک لیست زیر استفاده نماید.

عنوان		تنظیم صحیح	
		بله	خیر
۶	**تنظیمات رویدادنگاری**		
۶-۱	الگوریتم رمزنگاری کلید متقارن	☐	☐
۶-۲	طول کلید نامتقارن	☐	☐

۷ راهنمای ابزار مقاوم‌سازی

این ابزار دارای سه فایل اجرایی است که در ادامه به بررسی هر یک از آنها می‌پردازیم.

۷-۱ فایل start.sh

این فایل اسکریپت، تنها فایلی است که کاربر باید آن را اجرا کند. بقیه اسکریپتها از طریق این فایل اسکریپت فراخوانی و اجرا می‌شوند.

برای آنکه فرآیند تست سمپاد و امن‌سازی آن صورت گیرد، ابتدا لازم است از وجود MSSQL روی سیستم اطمینان حاصل کرد. در صورتی که سمپاد MSSQL بر روی سیستم نصب باشد، وجود بسته mssql-tools برای اتصال به سرور SQL و اجرای دستورات بررسی می‌شود. در صورتی که این بسته بر روی سیستم نصب نشده باشد، فرآیند نصب آن آغاز می‌شود. پس از آن، نام کاربری، رمز عبور و پورت برای اتصال به سرور SQL از کاربر دریافت می‌شود. بهتر است از نام کاربری SA برای اتصال و اجرای دستورات استفاده نشود و به جای آن کاربری با مجوز sysadmin به کارگرفته شود. حال اسکریپت script.sh اجرا می‌شود. در اثر اجرای این فایل، دو پوشه حاوی نتایج آزمایش و نتایج مورد انتظار ایجاد می‌شوند. حال در این اسکریپت قصد داریم این دو پوشه را با هم مقایسه کنیم تا مغایرت‌های سیستم با موارد امنیتی مورد انتظار مشخص شوند. نتایج این آزمایش در فایل first_test_result ثبت می‌شود. نمونه‌ای از خروجی این برنامه در شکل (۱) نشان داده شده است.

```
     "MSSQL RESULT"                                    "EXPECTED RESULT"

<------1-1-Status of SA Login Account ------>          <------1-1-Status of SA Login Account ------>
sa 0                                                  | NONE


<------1-2-SA Login Renamed ------>                   <------1-2-SA Login Renamed ------>
sa                                                    | NONE


<------1-3-SA Name Existence ------>                  <------1-3-SA Name Existence ------>
1 sa                                                  | NONE


<------2-1-Ad Hoc Distributed Queries Parameter ------>   <------2-1-Ad Hoc Distributed Queries Parameter ------>
Ad Hoc Distributed Queries 0 0                           Ad Hoc Distributed Queries 0 0


<------2-10-Port Parameter ------>                    <------2-10-Port Parameter ------>
1433                                                  | port = '<Appropriate value>'
                                                      > port != 1433


<------2-12-xp_cmdshell Parameter ------>             <------2-12-xp_cmdshell Parameter ------>
xp_cmdshell 0 0                                       xp_cmdshell 0 0
```

شکل ۱: محتوای فایل first_test_result

ستون سمت چپ نشان دهنده تنظیمات فعلی است. مواردی که با تنظیمات مورد انتظار مغایرت دارند در مقابل آنها نتیجه مورد انتظار آورده شده است. پس از اجرای این اسکریپت، در صورت تمایل کاربر، اسکریپت repair اجرا می‌شود. سپس هر مورد امنیتی که در آن نتیجه مورد انتظار و نتیجه حاصل از تست مغایر باشند، با موافقت

کاربر امن‌سازی شده و در آخر نیز تست مجددی بر روی سیستم انجام می‌شود. نتیجه تست دوم در فایل second_test_result ذخیره خواهد شد.

```
|     "MSSQL RESULT"                                          "EXPECTED RESULT"

<------1-1-Status of SA Login Account ------>                    <------1-1-Status of SA Login Account ------>
NONE                                                            NONE


<------1-2-SA Login Renamed ------>                             <------1-2-SA Login Renamed ------>
NONE                                                            NONE


<------1-3-SA Name Existence ------>                            <------1-3-SA Name Existence ------>
NONE                                                            NONE


<------2-1-Ad Hoc Distributed Queries Parameter ------>         <------2-1-Ad Hoc Distributed Queries Parameter ------>
Ad Hoc Distributed Queries 0 0                                  Ad Hoc Distributed Queries 0 0


<------2-10-Port Parameter ------>                               <------2-10-Port Parameter ------>
1477                                                          | port = '<Appropriate value>'
                                                              > port != 1433


<------2-12-xp_cmdshell Parameter ------>                       <------2-12-xp_cmdshell Parameter ------>
xp_cmdshell 0 0                                                 xp_cmdshell 0 0
```

شکل ۲: محتوای فایل second_file_result

۷-۲ فایل script.sh

در این فایل دو متغیر با نام‌های result_path و expected_path تعریف شده است. این دو متغیر به پوشه‌هایی اشاره می‌کنند که در آن‌ها به ترتیب نتایج هر آزمایش و نتیجه مورد انتظار آن آزمایش، ساخته می‌شود. در اینجا لازم است بنا به نیازمندی سیستم و نکات گفته شده حین توضیح هر مورد، نتایج مورد انتظار برای هر مورد امنیتی تنظیم شود. این کد توسط برنامه start.sh اجرا می‌شود.

۷-۳ فایل repair.sh

فایل آخر، مربوط به تغییر تنظیمات سیستم می شود. در صورتی که تنظیمات به درستی انجام شده باشد انتظار می‌رود که مورد امنیتی در ستون دوم وجود نداشته باشد و تنها چند پیشنهاد برای امنیت بیشتر در آن باقی بماند. در شکل (۲) می‌توان نمونه ای از فایل second_test-result را پس از اعمال تغییرات مشاهده کرد.

۸ جمع‌بندی

در این مستند به بررسی موارد امنیتی مربوط به مقاوم‌سازی MSSQL پرداخته شد. تنظیمات مربوط به مقاوم‌سازی MSSQL در شش فصل مختلف دسته‌بندی شدند. در فصل اول، امن‌سازی محیط اجرا، فصل دوم نصب و پیکربندی امن پایگاه‌داده، فصل سوم امن سازی اتصال به پایگاه‌داده، فصل چهارم تنظیمات کنترل

دسترسی و مجازشماری، فصل پنجم تنظیمات رویدادنگاری و فصل ششم تنظیمات رمزنگاری بررسی شدند. در مورد هر پارامتر، کاربرد، ارزش امنیتی و نحوه آگاهی از مقدار کنونی آن پارامتر و چگونگی مقداردهی امن آن توضیحاتی ارایه گردید. در پایان نیز نحوه اجرای اسکریپت‌ها و خروجی‌های سیستم بیان شد. خلاصه‌ای از گزارش ارائه شده به صورت چک لیست در ادامه آورده شده است.

	عنوان	تنظیم صحیح	
		بله	خیر
۱	**ایمن‌سازی محیط اجرا**		
۱-۱	پیکربندی فایل تنظیمات	☐	☐
۱-۲	پیکربندی دایرکتوری ذخیره داده	☐	☐
۱-۳	پیکربندی فایل‌های رویدادنگاری	☐	☐
۱-۴	غیرفعال کردن حساب کاربری sa	☐	☐
۱-۵	تغییر نام حساب کاربری sa	☐	☐
۱-۶	حذف حساب کاربری با نام sa	☐	☐
۲	**پیکربندی امن پایگاه‌داده**		
۲-۱	پارامتر Ad Hoc Distributed Queries	☐	☐
۲-۲	پارامتر CLR Enabled	☐	☐
۲-۳	پارامتر Cross DB Ownership Chaining	☐	☐
۲-۴	پارامتر Database Mail XPs	☐	☐
۲-۵	پارامتر Ole Automation Procedures	☐	☐
۲-۶	پارامتر Remote Access	☐	☐
۲-۷	پارامتر Remote Admin Connections	☐	☐
۲-۸	پارامتر Scan For Startup Procs	☐	☐
۲-۹	پارامتر Trustworthy	☐	☐
۲-۱۰	پارامتر TCP port	☐	☐
۲-۱۱	پارامتر Hide Instance	☐	☐

۲-۱۲	پارامتر xp_cmdshell	☐	☐
۲-۱۳	پارامتر AUTO_CLOSE	☐	☐
۲-۱۴	مجموعه مجوزهای اسمبلی CLR	☐	☐
۲-۱۵	سرویس پک و اصلاحیه‌ها	☐	☐
۳	**امن‌سازی اتصال به پایگاه‌داده**		
۳-۱	پارامتر Server Authentication	☐	☐
۳-۲	حذف کاربران یتیم	☐	☐
۳-۳	احراز اصالت در پایگاه‌های داده contained	☐	☐
۳-۴	لاگین با گروه‌های پیشفرض	☐	☐
۳-۵	لاگین با گروه‌های محلی ویندوز	☐	☐
۳-۶	پارامتر MUST_CHANGE	☐	☐
۳-۷	پارامتر CHECK_EXPIRATION	☐	☐
۳-۸	پارامتر CHECK_POLICY	☐	☐
۴	**کنترل دسترسی و مجازشماری**		
۴-۱	مجوز CONNECT	☐	☐
۴-۲	تنظیم حساب خدماتی سرویس MSSQL	☐	☐
۴-۳	تنظیم حساب خدماتی SQLAgent	☐	☐
۴-۴	تنظیم حساب خدماتی سرویس Full-Text	☐	☐
۴-۵	تنظیم مجوزهای نقش public	☐	☐
۴-۶	تنظیم دسترسی نقش public به پروکسی‌های SQL Agent	☐	☐
۵	**تنظیمات رویدادنگاری**		
۵-۱	تنظیم تعداد فایل‌های رویدادنگاری خطا	☐	☐
۵-۲	پارامتر Default Trace Enabled	☐	☐
۵-۳	پارامتر Login Auditing	☐	☐
۵-۴	تنظیم SQL Server Audit	☐	☐

۶	تنظیمات رمزنگاری		
۶-۱	الگوریتم رمزنگاری کلید متقارن	☐	☐
۶-۲	طول کلید نامتقارن	☐	☐

۹ مراجع

[1] https://www.cisecurity.org/

[2]https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-release-notes?view=sql-server-۲۰۱۷

[3] https://docs.microsoft.com/en-us/sql/t-sql/statements/create-user-transact-sql?view=sql-server-2017