MsSQL: Grundlæggende sikkerhed

En kort gennemgang af grundlæggende sikkerhed på SQL Server ved brugerkonti og opsætning.

Skrevet den **05. Feb 2009** af **trer** I kategorien **Databaser / MS SQL** | ★★★☆☆

Microsoft SQL Server (MsSQL) findes i en række forskellige varianter - MSDE (Microsoft SQL Server Desktop Edition), Personal Edition, Standard Edition, Enterprise Edition og DataCenter edition. Men uanset hvilken version man har installeret er der de samme grundlæggende sikkerhedsting man skal have på plads.

På Windows NT Serien af operativsystemer er MsSQL en service. Som en sådan skal den køre under en brugerkonto. Standard er, at MsSQL er installeret under Local System Account, men den kan køre under andre konti, blot disse har administrative privilegier i Windows.

Det at SQL Server har administrative rettigheder på operativsystemet gør, at såfremt man tiltvinger sig adgang til SQL Serveren, så har man gode chancer for at tiltvinge sig administrative rettigheder på selve operativsystemet.

Den administrative superkonto på en SQL Server hedder SA (forkortelse for System Administrator). Man bør altid sikre sig, at denne konto har et ekstremt langt og kompliceret password (se tippet nedenfor) og i stedet oprette sin egen konto som man tildeler administrative rettigheder.

Dette gøres således

```
use master
go
exec sp_addlogin 'john','minadgangskode'
exec sp_addsrvrolemember john,sysadmin
```

Når SA kontoen nu ikke længere skal benyttes, så kan man gøre den temmelig ubrugelig ved at benytte følgende lille tip:

Indsæt et semikolon (;) et sted i adgangskoden. Således:

```
use master
go
exec sp_password @loginame='sa',@new='eu3bask3;9lkak4ek'
```

Fidusen er, at et semikolon fortæller at et nyt felt i ens connectionstring starter - og dermed er adgangskoden aldrig korrekt. Man bør dog ikke løbe an på det og nøjes med at benytte et semikolon. Adgangskoden bør stadig være en tilfældig række af bogstaver og tal, gerne omkring 15-20 tegn. Længden vil jo ikke genere når man alligevel ikke skal bruge kontoen.

På SQL Server findes i master databasen en række extended stored procedures - typisk er de navngivet i formatet "xp_xxxxx", men enkelte af dem er navngivnet "sp_xxxxx".

Nemmeste måde at finde listen på, er ved at lave følgende query i master-databasen:

```
select name
from master.dbo.sysobjects
where xtype='x'
```

Disse extended stored procedures kalder funktioner i dll'er i operativsystemet, og giver bl.a. mulighed for at afvikle kommandoer.

fx vil kommandoen

```
xp_cmdshell 'net localgroup administrators /add john'
```

tilføje brugeren john til den lokale administrator-gruppe...

Man bør derfor som standard altid sikre, at ingen normale brugere har ret til at afvikle extended stored procedures. Nemmeste måde at sikre dette på, er
ved at bruge DENY EXECUTE til PUBLIC. Man kan lave et hurtigt og nemt script til at sætte disse rettigheder således

```
select 'DENY EXECUTE ON '+name+' TO PUBLIC'
from master.dbo.sysobjects
where xtype='x'
```

Outputtet fra dette er en række SQL Statements man kan fyre af mod serveren, derefter vil kun DBO i master-databasen samt folk med administrative rettigheder have mulighed for at afvikle extended stored procedures.

En anden gruppe af procedurer der kan give problemer er job-procedurerne. Dette er normale stored procedures, men de benyttes til at sætte jobs op og
her i ligger en risiko. Et job har nemlig en filbaseret log-funktion. Man kan i definitionen af jobbet placere logfilen hvor man ønsker det, og man kan navngive filen som man ønsker.

Det er dermed muligt at lave en log-fil i fx "C:\Documents and Settings\All Users\Start Menu\Programs\Startup". Kald logfilen et eller andet med .bat, og man kan lave et script der afvikles når næste bruger logger på - eller logfilen kan placeres så den overskriver operativsystemets filer...

Løsningen er som ovenfor - nægt ret til at almindelige brugere kan definere jobs og starte dem.

```
select 'DENY EXECUTE ON '+name+' TO PUBLIC'
from master.dbo.sysobjects
where xtype='p'
and name like '%_job%'
```

Ovenstående forhindrer altså kun almindelige brugere i at afvikle disse uønskede procedurer - folk med DBO ret eller administrative rettigheder kan stadig afvikle dem.

Man bør sørge for, at der ALDRIG benyttes andet end almindelige brugere til at tilgå databasen med. DBO kontoen bør specielt undgås i almindelige databaser da denne konto har hånd og halsret over alt i den enkelte database - inklusive mulighed for at droppe den! En god ting er, at give DBO-konti samme type adgangskoder som ovenfor beskrevet ved SA.

I et veldesignet system - specielt til web-brug - er det nok at definere en række stored procedurer i hver

database. Brugerkonti skal så kun have adgang til at afvikle procedure, ikke til at tilgå tabeller, views etc.

Man har dermed defineret præcis hvilke operationer der er mulige at foretage i databaserne, og da al tilgang til data sker via disse procedurer så er de de mest almindelige former for SQL Injection ikke længere mulige.

En script til at grante brugeren JOHN ret til at afvikle ens egne procedurer kan laves således:

```
select 'GRANT EXECUTE ON '+name+' TO john'
from dbo.sysobjects
where xtype='p'
and objectproperty(id,IsMsShipped)=0
```

I øvrigt, man bør altid oprette roller og tildele rettigheder til disse fremfor at tildele rettigheder til enkelt brugere. Ens database bliver væsentligt nemmere at administrere på den måde.

En anden ting er, at administrative konti på operativsystemet som standard har administrative rettigheder i SQL Serveren. Sikkerhedsmæssigt lyder det måske ikke som det store problem, men hvis nogen tiltvinger sig adgang via en anden applikation eller service, så har de nu også fulde rettigheder til SQL Serveren.

En simpel løsning er, at fjerne gruppen BUILTIN\Administrators fra SQL Server eller måske simpelthen at fjerne gruppens tilknytning til SysAdmin gruppen. Dermed er medlemmer af administratorgruppen i Windows ikke længere automatisk system administratorer i SQL Server.

Man skal lige være opmærksom på, at så andre services - f.eks. SQL Server Agent Servicen - kan få problemer med at forbinde til SQL Server når man forhøjer sikkerheden. Benytter man ikke Agent Servicen er det ikke noget problem, og ellers må man blot oprette en brugerkonto med de fornødne rettigheder til den pågældende service. Og igen - sørg for at lave en ekstremt lang og kompleks adgangskode.

Overholdes disse ting - og har man i øvrigt patchet sin installation op - vil en eventuel hacker/crackers arbejde være rimelig besværliggjort.

Som sidste detalje kan man så passende sætte auditlevel på SQL Server til at logge alle fejlslagne forsøg på at logge på. Kig loggen gennem fx uge eller månedligt og se om nogen forsøger sig gentagne gange.

Auditering sættes nemmest via Enterprise Manager ved at højreklikke på serveren, vælge properties og gå ind på fanen Security. Her sætter man Audit Level til enten full eller Failure og trykker OK.

God fornøjelse
Troels

Kommentar af mercur8 (nedlagt brugerprofil) d. 27. Jan 2004 | 1

SQL Server bør indgå i artiklens titel. Kan ikke bedømme indholdet.

Kommentar af bufferzone d. 25. Jan 2004 | 2

Kort og præcis artikel, der bør læses af alle der interessere sig for IT sikkerhed, og især for folk der her en SQL server i drift. Til pennen Trols!!! flere artikler!!!

Kommentar af bennytordrup d. 13. Feb 2004 | 3

Udmærket artikel.

Kommentar af zedios d. 29. Jan 2004 | 4

Kommentar af squashguy d. 30. Jan 2004 | 5

Kommentar af pct d. 09. Mar 2004 | 6

Kommentar af dbay d. 05. Jan 2005 | 7