Wachtwoorden en encryptie

Oracle en security (9)

Voor veel geautomatiseerde systemen vormen wachtwoorden de enige barrière voor de toegang tot data. Bij de meeste organisaties waar één of meerdere Oracle-databases gebruikt worden, is het voor veel medewerkers mogelijk om via tools als SQL*Plus of TOAD toegang te verkrijgen tot deze databases. In dit artikel zal Gerard Uiterwaal een aantal van de veiligheidsrisico's bespreken die het standaardgebruik van wachtwoorden in een Oracle database met zich meebrengt.

In de Oracle-database worden wachtwoorden in het algemeen niet onversleuteld opgeslagen. Ze worden versleuteld met een onomkeerbaar proces met als resultaat een *password-hash*. Onomkeerbaar betekent in dit geval, dat het onmogelijk is om uit de password-hash het wachtwoord terug te herleiden. Dit proces heeft als invoer de gebruikersnaam en het wachtwoord en levert dan altijd dezelfde password-hash op. Bij het creëren van een user wordt dit proces gebruikt om deze hash te bepalen en op te slaan in de tabel USER$ van SYS in de dictionary. Bij het aanloggen van een user wordt de hash opnieuw bepaald en vergeleken met die uit de tabel USER$. Indien de hashes gelijk zijn, gaat Oracle ervan uit dat het gebruikte wachtwoord correct is.

Zwakheden

Het algoritme dat Oracle gebruikt voor de versleuteling van wachtwoorden is een sterk algoritme. Het is bij mijn weten nog nooit direct gekraakt. Wel heeft de procedure een ingebouwde zwakte:

- Voor elke database geldt dat voor eenzelfde user/wachtwoord-combinatie eenzelfde password-hash oplevert.

Wanneer je ziet dat de password-hash van SYS gelijk is aan D4C5016086B2DC6A of dat van SYSTEM gelijk is aan D4DF7931AB130E37, dan weet je dat het wachtwoord van SYS het bekende CHANGE_ON_INSTALL is, of het wachtwoord van SYSTEM het bekende MANAGER.

Het gevaar van deze password-hashes in combinatie met de bijbehorende gebruikersnaam is dat je volledig buiten de database om kan gaan uitproberen wat het bijbehorende wachtwoord is. Je kunt ze vergelijken met een lijst van bekende password-hashes, maar je kunt ook password-hashes genereren en ze één voor één vergelijken. Iemand die probeert om in een database binnen te komen, kan op deze manier auditing omzeilen. Hij werkt namelijk niet in de database waar hij binnen probeert te komen, tot het moment dat hij de juiste user/wachtwoord-combinatie kent.

Een tweede zwakheid is dat er toch plekken zijn waar onversleutelde wachtwoorden opgeslagen worden. Met name het wachtwoord van een database-link wordt onversleuteld opgeslagen. Afhankelijk van de versie van Oracle is het te vinden in de tabel LINK$ van SYS en soms ook in de view DBA_DB_LINKS.

Een derde zwakheid is het gebruik binnen Oracle van een groot aantal default-accounts met bijbehorende bekende wachtwoorden. Het is iedere Oracle-gebruiker bekend dat de standaard DBA-account SYSTEM heet en de meeste weten ook dat het default-wachtwoord MANAGER is. Ook het SYS-account kennen de meeste nog wel met zijn standaardwachtwoord CHANGE_ON_INSTALL. Dat een standaard-geïnstalleerde database nog wel tachtig andere accounts kan meekrijgen met eveneens bekende wachtwoorden is echter veel minder bekend. Eén van de meest vervelende hiervan is de account DBSNMP met als wachtwoord DBSNMP en vrij veel privileges. Onbekend bij zelfs de meeste DBA's is dat bij het installeren van omvangrijke patches dit account opnieuw wordt aangemaakt en wel met zijn standaardwachtwoord. Pas vanaf Oracle

Zorg ervoor dat de toegang tot password-hashes zo veel mogelijk beperkt wordt

9i is Oracle ermee begonnen om deze accounts standaard beter dicht te zetten.

Mogelijke maatregelen

Zorg ervoor dat de toegang tot password-hashes zo veel mogelijk beperkt wordt.

Deze hashes staan op een aantal plaatsen:
- Dictionary tabellen en views (de tabel USER$ van SYS en de hierop gebaseerde views zoals DBA_USERS, EXU7ROL, EXU8ROL, EXU8PHS, KV$_USER_VIEW en KV$_ROLE_VIEW)
- De password file

De toegang tot de dictionary wordt tot Oracle 8 voornamelijk geregeld via het *select any table privilege*, dat bijvoorbeeld in de DBA-rol zit, maar ook individueel kan worden toegekend. Vanaf Oracle 9 is dit het *select any dictionary privilege* geworden. Het uitreiken van deze privileges moet zoveel mogelijk vermeden worden. De password-file wordt gebruikt om via OS-groepen te kunnen bepalen of een user mag aanloggen en met welke privileges (OSDBA of OSOPER). Alle gebruikers waaraan één van deze rollen wordt toegekend, worden met password-hash in de password-file opgenomen.

Bekijk de lijst met accounts kritisch en verwijder degene, die niet gebruikt worden

Bekijk de lijst met accounts kritisch en verwijder degene, die niet gebruikt worden.
Let hierbij op bij het verwijderen van door Oracle meegeleverde accounts. DBSNMP bijvoorbeeld wordt door de intelligent-agent gebruikt. Wel is het mogelijk het wachtwoord van dit account aan te passen.

Zorg ervoor dat wachtwoorden niet te gemakkelijk te raden zijn, zeker niet voor accounts met veel privileges.
Gebruik geen bestaande woorden als wachtwoord, al dan niet aangevuld met een postfix of prefix (bijvoorbeeld 'woord' of 'woord01').

Beperk de default toegang tot de view ALL_USERS.
De view ALL_USERS bevat weliswaar geen password-hashes, maar is een bron van informatie om achter gebruiker/wachtwoord-informatie te komen. Er zijn meestal wel gebruikers met zwakke wachtwoorden en sommige hebben meer privileges dan ze zich realiseren. Alhoewel PUBLIC select rechten heeft op de view ALL_USERS (zoals op alle dictionary views met de prefix ALL_), is het mogelijk en wenselijk om dit select recht van PUBLIC af te nemen.

Bekijk periodiek en zekere na het aanbrengen van een patch de lijst met bekende accounts.

Wachtwoord in netwerkverkeer

Het wachtwoord in de aanlogprocedure neemt in het netwerkverkeer een aparte plaats in. Al sinds Oracle 7.1 wordt het wachtwoord gedurende de aanlogprocedure altijd versleuteld. Om ervoor te zorgen dat nieuwere clients aan pre-7.1 databases kunnen aanloggen zijn er een tweetal initialisatieparameters toegevoegd:

- `ora_encrypt_login`
- `dblink_encrypt_login`

Indien deze de (default)-waarde FALSE hebben, wordt er indien het aanloggen met een versleuteld wachtwoord niet lukt, een tweede poging gedaan met het onversleutelde wachtwoord. Vanaf versie 9.2 zijn deze parameters niet meer zinvol, aangezien een 9.2 client niet meer in een 7-database kan aanloggen. Vanaf 10g zijn de parameters dan ook *obsolete*. Indien op een site alleen nog maar databases gebruikt worden met versie 7.1 of hoger is het een veiligheidsrisico om deze parameters op de default-waarde te laten staan. Dit is niet zozeer omdat er goede wachtwoorden onversleuteld over het netwerk zullen gaan. Deze gaan versleuteld over het netwerk en de poging om aan te loggen lukt. Voor een bijna-goed wachtwoord geldt dit echter niet. De aanlogpoging met de versleutelde versie mislukt en het bijna-goede wachtwoord gaat alsnog onversleuteld over het netwerk voor een nieuwe poging, die natuurlijk weer mislukt. Voor de encryptie van het netwerkverkeer is, met uitzondering van het wachtwoord in de aanlogprocedure zoals hiervoor beschreven, de Advanced Security-optie nodig.

Oracle ondersteunt ook een tweetal data-integriteit algoritmes:
- Message Digest 5 (MD5)
- Secure Hash Algorithm (SHA-1)

In tegenstelling tot de bovengenoemde encryptie-algoritmes, die symmetrisch zijn, zijn deze *one-way*. In dit opzicht zijn ze vergelijkbaar met het algoritme dat Oracle gebruikt voor de versleuteling van zijn wachtwoorden. In veel Linux-smaken wordt MD5 dan ook gebruikt voor de versleuteling van wachtwoorden.

Binnen Oracle worden deze algoritmes gebruikt voor een heel ander doel. Namelijk om te kunnen vaststellen of een bericht onderweg verminkt is. Door zowel voor verzending als na verzending een hash te bepalen en deze met elkaar te vergelijken kan worden vastgesteld of hetzelfde bericht is aangekomen als

is verzonden. Ook het meermaals herhalen van één bericht kan hiermee worden voorkomen.
Het doel van versleuteling is geheimhouding, terwijl het doel van deze twee algoritmes is om de integriteit van het bericht te bewaken. Binnen Oracle kan er dan ook voor gekozen worden om één van deze technieken te gebruiken, om ze allebei te gebruiken of geen van beide.

Misbruik van password-hash

Password-hashes worden ook gebruikt (of misbruikt) om aan te kunnen loggen aan de database met een account waarvan het wachtwoord niet bekend is. Om onderstaande te kunnen uitvoeren zijn er twee voorwaarden:
• De password-hash moet toegankelijk zijn, bijvoorbeeld via de view DBA_USERS.
• Je moet beschikken over het alter user privilege.
Normaliter is dit alleen mogelijk voor gebruikers met de DBA-rol.

De procedure werkt als volgt:
• Zoek de password-hash op van de gebruiker waarmee je wilt aan loggen, bijvoorbeeld GEB1 en de password-hash HASH1
• Wijzig het wachtwoord van deze gebruiker

```
alter user GEB1 identified by GEHEIM;
```

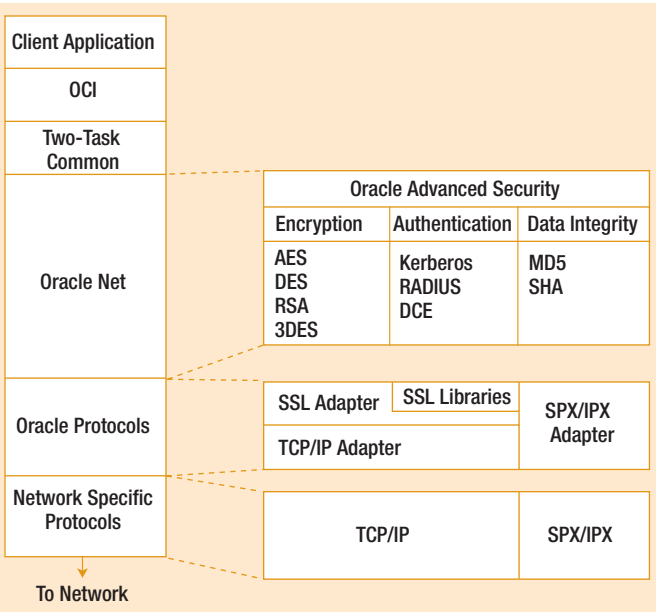
• Log aan als deze gebruiker

```
connect GEB1/GEHEIM
```

• Wijzig als gebruiker GEB1 je eigen wachtwoord (hier heb je geen privileges voor nodig)

```
alter user GEB1 identified by values 'HASH1';
```

Het geheim van de procedure zit hem in het gebruik van het keyword-values. Hierdoor wordt de waarde van het opgegeven wachtwoord zonder versleuteling opgeslagen en wordt de oude password-hash weer teruggezet. Het resultaat is dat de gebruiker zijn oude wachtwoord weer terug heeft, overigens zonder dat de DBA in kwestie dit wachtwoord kent. Een andere manier waarop deze syntax misbruikt kan worden, is door een gebruiker een password-hash te geven, waar geen wachtwoord bij hoort. Er zijn tegenwoordig echter nettere en betere methodes om accounts dicht te zetten.



Figuur 1. Voor encryptie van het netwerkverkeer is de Advanced Security-optie nodig.

Gerard Uiterwaal is Oracle- en security expert en werkzaam bij Motiv. Indien u in een van de volgende uitgaven van Optimize graag een specifiek onderdeel belicht zou willen zien dan kunt u dit aangeven via e-mail: gerard.uiterwaal@motiv.nl . De auteur streeft ernaar zoveel mogelijk vragen te beantwoorden.