



Passwords In Oracle

Slavik Markovich
CTO, Sentrigo

About Me

- Co-founder and CTO of Sentrigo
- Frequent presenter in Oracle and Security conferences
- DBA since 1996
- Creator of FuzzOr - a free Oracle Fuzzer
- <http://www.slaviks-blog.com>



Agenda

- ◆ Different password algorithms
- ◆ Password storage
- ◆ Database password algorithms
- ◆ Choosing passwords
- ◆ Countermeasures



Different Password Algorithms

- Hashing is a one-way method to convert a value into a hash value. Decrypting is not possible.
- Encryption is using a key to convert the plain text to encrypted text. It is possible to decrypt the encrypted string using the key.
- Passwords used for authentication (e.g. verified during login) are normally hashed (before transmitted over the network). Password which are used to connect to a system (e.g. passwords in Grid Control), job scheduling systems, passwords stored in database clients) are normally encrypted and can be decrypted. Often with a simple SQL statement (select decrypt(password) from tablepw)



Password Storage

- The database (e.g. tables, PL/SQL Code, ...)
- The memory of the database (e.g. v\$sql, bind parameter)
- The file system of the database server (e.g. dads.conf/marvel.conf)
- The file system of the application server (e.g. oc4j.conf)
- On the client(s) (e.g. connections.ini)



Passwords In Tables

- SYS.USER\$ (hashed: Oracle PW Alg)
- SYS.USER_HISTORY\$ (hashed: Oracle PW Alg)
- SYS.LINK\$ (since 10.2 encrypted)
- SYS.WRH\$_SQLSTAT (sometime SQL statements contains pw info)
- SYS.AUD\$ (some SQL statements contains pw info)
- Custom plsql-code
- Custom tables (e.g. %CRED% or %PASSW% or PWD)
- Oracle HTMLDB/APEX-Table (hashed: MD5, since 3.x salted MD5)
- OID: MD4, MD5, SHA-1
- OVS: MD5
- Various tables from Oracle products (Secure Enterprise Search, Oracle Lite, OMS, Peoplesoft, ...)
- Oracle database & products store password information in more than 100 different tables.



Passwords In Memory

- There are more than 150 function/procedure calls in Oracle accepting clear text passwords. If such a function is called the passwords are visible for a (limited time) in the database in memory (SGA) via the views v\$sql, v\$sql_area or if bind variables were used in v\$sql_bind_capture.

```
SQL> exec owa.set_password('superduper');
```

PL/SQL procedure successfully completed.

```
SQL> select sql_text from v$sql where lower(sql_text) like  
      '%owa.set_passw%';
```

```
select sql_text from v$sql where lower(sql_text) like 'owa.set_passw%'  
BEGIN owa.set_password('superduper'); END;
```

- Check DBA_ARGUMENTS



Passwords In Packages

```
my_private_encryption.sql (~/Oracle/Security) - VIM
File Edit View Terminal Tabs Help
CREATE OR REPLACE PACKAGE my_private_encryption IS
  FUNCTION encrypt(p_clear_text IN VARCHAR2) RETURN VARCHAR2;
  FUNCTION decrypt(p_cipher IN VARCHAR2) RETURN VARCHAR2;
END my_private_encryption;
/

CREATE OR REPLACE PACKAGE BODY my_private_encryption IS
  g_key      RAW(128)      :=      utl_raw.cast_to_raw('secret99');

  FUNCTION encrypt(p_clear_text IN VARCHAR2) RETURN VARCHAR2 IS
  BEGIN
    RETURN RAWTOHEX(dbms_crypto.encrypt(
      src => utl_raw.cast_to_raw(p_clear_text),
      typ => dbms_crypto.des_cbc_pkcs5,
      key => g_key));
  END encrypt;

  FUNCTION decrypt(p_cipher IN VARCHAR2) RETURN VARCHAR2 IS
  BEGIN
    RETURN utl_raw.cast_to_varchar2(dbms_crypto.DECRYPT(
      src => hextoraw(p_cipher),
      typ => dbms_crypto.des_cbc_pkcs5,
      key => g_key));
  END decrypt;
END my_private_encryption;
/
```

1,1 Top

wrap iname=... oname=...

```
my_private_encryption.plb (~/.Oracle/Security) - VIM
File Edit View Terminal Tabs Help
CREATE OR REPLACE PACKAGE BODY my_private_encryption wrapped
a000000
1
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
b
299 199
hGE6T3Sn140PB5Lg5NZ0x3EQSDUwggw3I1ydqfC9GEjNefW/98so35l63rFYgee6i05t79K5R
NizAEatxXoUF6puuqwcZCX1BK2qKJzNeiQEh+9P6UxQgF8flKE9kLgHlhrwiT+xq+Pela09h
07c1oUUL9R5yi9A3Hk+nkCuPl1QQLxFK0Wmec/qADWQKyS11NmuxSEI+NoTRRPuv9SZQ6kA7
qSfWS4QVRDFp1wnlwfuzqQL5b0B28xRxMJFaJVikFzZetDpSoP8umJPCB5bRyeq+yZE65AuD
HhavcxGeDEPo4sTGcAXjiI0t5DiKw7YmGxvtNmQ0WYcSLzp82pER0o+fZ0LU+0nBJvGr6rG
4PNRip0gk/kChE/K0mgLTKtj0kBmdUHQAdMZEPumiIz/
26,1 92%
```

Passwords In Files On DB Server

- listener.ora (hashed: Oracle PW Alg)
- Archive logs (hashed: Oracle PW Alg)
- Export files (hashed: Oracle PW Alg)
- Database Dump files (cleartext or hashed)
- Data files (hashed: Oracle PW Alg)
- Flash_recovery_area (Online_log) (hashed: Oracle PW Alg)
- offline_dictionary.ir (logminer 11g, hashed: Oracle PW Alg)
- Oracle password file (hashed: Oracle PW Alg)
- .htaccess (encrypted: CRYPT)
- wdbsvr.app, dads.conf , wfmail.cfg (cleartext, BASE64, encrypted)
- Oracle Installation files (cleartext)
- Oracle Trace files (cleartext)

Passwords In Application Server

- Java config files (cleartext)
- cgicmd.dat (cleartext)
- .htaccess (encrypted: CRYPT)
- Configuration files for database connection
- Hardcoded in binaries (e.g. ODSCOMMON in iAS 9.0.2)
- Installation files (cleartext)
- Trace files (cleartext)



Passwords In Registry

- ODBC
- Oracle Client Tools
- Oracle Apps



Oracle Password Algorithm

- Until 11g
 - Passwords up to 30 chars long and converted to uppercase
 - 8-byte hash, encrypted with a DES without salt
 - The Oracle password algorithm can be found in newsgroups or as plug-in (source) for JohnTheRipper
 - `scott/tiger == scottt/iger`
- 11g
 - New (optional) password algorithm
 - SHA-1 (password| |salt)
 - Password hash no longer visible in `dba_users` (get PW hash: `select name,spare4 from sys.user$`)
 - Enable Case-Sensitivity - `ALTER SYSTEM SET SEC_CASE_SENSITIVE_LOGON = TRUE`



Getting The Password Hash

- <11g

```
SELECT username, password FROM dba_users
```

- 11g

```
SELECT name, password, spare4 FROM  
sys.user$
```

- To avoid rootkits always select from user\$ and flush shared pool before if possible

Default Passwords

- Lists are available from various sites on the web like

http://www.petefinnigan.com/default/default_password_list.htm

- 11g lists ~700 default passwords in `sys.default_pwd$`
- `dba_users_with_defpwd`
- Oracle default password scanner available from Metalink



Password Attacks

- Intercept Password (hash) on the network (e.g. Wireshark)
- Watching the keyboard (e.g. shoulder surfing, camera)
- Keylogger (e.g. software, USB, PS/2 or built into the keyboard)
- Brute force attack (e.g. with woraaauthbf)
- Dictionary attack (e.g. with checkpwd or repscan)
- Rainbow Table attack (e.g. with ophcrack or cain)
- Dictionary based rainbow table attack (e.g. repscan or ophcrack)
- Authentication attack (e.g. with woraaauthbf or orakel)



New Cracking Tools Use Graphics Card

- In 2008 using the graphic card to crack passwords became more and more popular.
- There are 2 different framework available.
 - CUDA from NVIDIA
 - CTM/AMD Stream from AMD
 - CUDA is easier and more popular
- OpenCL will sooner or later replace these proprietary technologies



Oracle Password Checking Tools

- Checkpwd / Repscan from Red-Database-Security GmbH (smartest and most convenient tools, platform independent)
- Woraauthbff from Laszlo Toth (fastest tool for brute force/dictionary mode on Windows)
- Cain from Mao (using rainbow tables)
- PL/SQL Oracle Password cracker from Pete Finnigan
- Perl Oracle Password cracker from Alun Jones
- JohnTheRipper with Oracle Password patch from Solar Designer

Brute Force Attacks

- woraauthbf from Laszlo Toth is currently the fastest Oracle DES password cracker for Windows.
- Woraauthbf is open source but only available on Windows
- Performance:(4.4M PW per second on a 2.5 GHz Core2Quad) needs the following time to calculate all passwords in BF mode. Special hardware can do this up to 10,000 times faster...
- Checking random passwords is not the best way
- With CUDA 8 character passwords are breakable within days



Dictionary Attacks

- Repscan and Checkpwd from Red-Database-Security
- Can be easily scheduled to run periodically on all databases



Authentication Attacks - 10g

- The client sends the username and receives the AUTH_SESSKEY and decrypts it with ztvo5kd function. It uses the Oracle password hash
- Then the client calls the ztvo5kcs to combine the decrypted AUTH_SESSKEY from the server and a generated key. The two keys are XORed and the final key will be the MD5 hash of the XOR result.
- Then the client calls ztvo5ke to encrypt its generated key with password hash. The result will be sent as the AUTH_SESSKEY of the client.
- The next step is the password encryption with the result of the ztvo5kcs (because of the MD5 it is 128bit long). The called function is the ztvo5pe.
- The server receives the AUTH_SESSKEY of the client and the AUTH_PASSWORD.
- The AUTH_SESSKEY of the client is decrypted using the password hash with the ztvo5kd function.
- Then the server combines the decrypted value with its generated key (decrypted AUTH_SESSKEY of the server) with ztvo5kcs.
- With the result, it decrypts the AUTH_PASSWORD. (ztvotpd)



Passwords In Foreign Languages

haslo = polish

mima = chinese

parola = russian

sifre = turkey

salasana = finnish

jelszo = hungaria

mot de passe = french

khufia = hindi

clave = spanish

senha = portugese

lozinka = croatian

■ wachtwoord = dutch

wagword = africaan

slösenord = swedish

fjallkalim = albanian

parool = estonian

sisma = hebrew

sandi = indonesian

parole = latvian

geslo = slovene

Choosing Passwords

- Oracle Passwords are often identical for many databases
- DBAs have the problem to choose passwords for several different databases
- At least 4 passwords per database (SYS, SYSTEM, OUTLN and DBSNMP) must be chosen
- Nobody can remember hundreds of different and good passwords
- Most DBAs are using the same password for ALL databases. If you have 1 password, you have access to all databases



Choosing Passwords

- Common Approaches for Oracle Databases
 - Choose the same password for every database
 - Use a password schema using a prefix/postfix
P=production, T=test, E=education (e.g Tpassword)
 - Append the SID(e.g. Passwordora902)
 - Use the computer name (e.g. passwordUNIX04)
- Check password strength
 - <http://www.securitystats.com/tools/password.php>



Best Practices

- Clear history files on a regular basis
- Do not use passwords in the environment
- Avoid clear text passwords in configuration files
- Password must be 8 or more characters
- Use salted SHA1 for hashes
- Encrypt with salted 3DES / AES in DB
- Check for default / weak passwords periodically



Questions?

