



Defence

# Introduction to Oracle Security

Mikoláš Panský



Difficulty



**This Article is focused on Oracle Database Server Security. It is divided in three main parts. The First is about Oracle history, database products and Architecture. The Second part is about basic methods of Oracle Hacking. The last part is about Oracle Defense methods.**

Oracle Corporation history starts in 1977 when the company was founded as Software Development Laboratories. In 1979 was SDL renamed to Relation Software, Inc. (RSI). In this year the Company released Oracle v2 as one of the first commercial Relational Database System. This version implemented basic SQL functions: query and joins. Oracle Corporation has this name since 1983 and released version 3 written in C and supported transactions. In 1984 there were version four, 1985 version five (client-server model), 1989 Oracle Corp. entered the Application market with Oracle Financial and implemented PL/SQL. In 1992 there was version 7h – data Warehouse with the relational integrity support, stored procedures and triggers. In Year 1997 was developed version 8, that support object-orientated approach and multimedia applications, 1999 was released the version 8i with support of Internet and Java Virtual Machine (JVM). Year 2001 bring Oracle 9i with the possibility of reading XML documents, RAC (Real Application Clusters) support. Today, the actual version is 10g Release 2 with the Grid support.

Oracle is released in different versions. Each has different implemented features. This

Article is focused on Oracle Dataset. Oracle Database has several editions: Standard Edition (SE allow maximum 4 CPU, with no memory limit and it's usable in Cluster), Enterprise Edition (EE) includes some Advanced Security Functions. It's possible to add Database Vault, that allows data protection against Database Administrators (DBA), Advanced Security allows the network communication encryption, encryption of the data in database, stronger authentication and finally Label Security that allows the security privileges definition and user's label – the security on the row-level. Along this there si also Standard Edition One with support of maximum 2 CPU, Personal Edition

## You will learn from this Article

- General information about Oracle,
- Basic Hacking Oracle method,
- Basic Oracle Defence methods.

## What You should already know

- Basic knowledge of Oracle Database System.

without RAC targeted on developers and Express Edition with the 1 CPU, 1GB RAM and 4 GB data limit.

Oracle Database System from the physical view is composed of processes, that runs on host operating system, logical memory structure (Instance) and physical file structure – Database. Processes are divided on user processes and server processes. Anytime user runs application, user process connects to Instance. If the Communication is established, the Session is started. For each user Server allocated PGA (Program Global Area) where it stores session variables. Oracle Instance is made by main memory structure SGA (System Global Area) and processes that runs on background. The most important processes are System Monitor – SMON (takes care about recovering from disaster, compacts free space in Database), Process Monitor – PMON (monitoring running processes and ensure it's support), DBW – Database Writer and Log

Writer – LGWR (writes the records that allows roll back). Oracle Database is composed of Control Files (control files that includes Database name, Data files placement and Redo Logs), Data Files and Redo Logs (that records all changes in Database). Information of the running processes are placed in tables V\$PROCESS and V\$SESSION. The communication with outer world is handled by Oracle Listener. It's configuration is sorted in the listener.ora file. SID (Oracle System Identifier, that resolves database Instance and identify database), protocol and port are stored in listener.ora. Listener listens for database requests. After receiving any connection, it sends TCP port number to the client. Client then connects to the port and authenticates itself. Listener could be also used by PL/SQL package or external procedures.

The Logical Database Structure is composed of users, schemas (objects owned by user), rights, roles,

profiles and objects. Users in the Database are Unique identities, that has access to the Database Objects. Users are most frequently identified by password. Each user has Schema, which is owned by him and where his objects are stored. Privileges is set of operations, that could User use. Profiles si set of options that restricts Database usage. It could define maximum retries of entering password before the account will lock down etc. Tables has rows and column. Access to the tables can be defined and restricted on the row basis with Virtual Private Database. Triggers are stored programs, that runs on event like inserting into table or shutting down database. Stored procedures are programs written in PL/SQL (*Programming Language SQL*). All information about Database are stored in Data Dictionary.

## Hacking Oracle

Before we will begin, there must be preceding phase of target network exploring. This Phase has to research detailed information, that could be retrieved by the Whois database, Internet Search Engines, DNS Servers or by Social Engineering. Search Engine could be also used to find required system according to search string, that is unique identifier for the right page. This search string could for example look for isqlplus (web interface for entering queries to Oracle Database), configuration files or Express Edition. The search strings could look like: *intitle:icql intitle:release inurl:isqlplus, listener filetype:ora* či *inurl:apex intitle:Application Express Login*. The next step is further scan of the operating system. This could be done by active tools (nmap, amap, tsnping) or passive (scanrad). The basic thing to do is to scanning open ports. Oracle in the standard configuration listens on standard ports that could be identified. To find running Listener could be used tool TSNPING. After the Database Server was found, we could try to obtain Version, Platform, SID and configuration. Tool to do this TSNLSNR IP Client, that could

### Listing 1. New Profile Creating

```
CREATE PROFILE paranoid LIMIT
  FAILED_LOGIN_ATTEMPTS 3
  PASSWORD_LOCK_TIME 30
  PASSWORD_LIFE_TIME 90
  PASSWORD_GRACE_TIME 3
  PASSWORD_VERIFY_FUNCTION check_the_password;
```

### Listing 2. Example of Function that could check the password

```
CREATE OR REPLACE FUNCTION check_the_password
(i_am_user_id VARCHAR2, new_magic_word VARCHAR2, old_magic_word VARCHAR2)
RETURN BOOLEAN IS
BEGIN
  IF length(new_magic_word) < 5 THEN
    raise_application_error(-20001, 'Your Magic Word Is Too Short!');
  END IF;
  IF NLS_LOWER(new_magic_word) IN ('password', 'drowssap') THEN
    raise_application_error(-20002, 'I will Not Accept Your Magic Word');
  END IF;
  RETURN TRUE;
END;
```

### Listing 3. Function that returns string which will be added to the query

```
CREATE OR REPLACE FUNCTION deny_table_rows (
  usr_schema VARCHAR2,
  usr_object VARCHAR2) RETURN VARCHAR2 AS
BEGIN
  RETURN 'user != SYS';
END;
```



provide commands ping, version, service and status. Requested information could be obtained only while Administrator didn't set password for Oracle Listener. If the password is set the Listener cannot be used for obtaining information. There is more tools available for Listener exploring: TNSCmd and OScanner. Commercial products, that could be used for this purpose is NGSSQuirrel. This is quite complex program and has many features. Some of them are available only with Oracle account, however it could also provide Dictionary or Brute Force attack on the user's accounts. If there is non-secured Listener, several possibility to attack could be done. In past, there were several security alerts. Some of them are *NERP* DoS attack, too large segment attack, illegal version request, too small size of transferred data, Fragmentation Attack

or *SERVICE\_NAME* DoS attack. Exept these it is possible to change Listener password that results in HiJacking, stopping the Listener or parameters change with SET command. If there was SID found and we know the version there is time to try some user name and passwords. The first should be to try the same user's names and passwords. Next we could try default User names and passwords. The next in the row would be dictionary and finally brute force attack. To check the user names and passwords tool called Hydra could be used.

The next possibility, how to obtain access on the database server Oracle is to sniff the connection. If the communication between user and client is unsecured, it could be sniffed by any network sniffer. At first user sends user name to the database. If the user name exists then

the server checks user's password hash. It uses secret number that is composed on the system time.

After obtaining the access to the database, it is necessary to check, if it's possible to escalate the rights for working with the system. The most common methods are SQL Injections, Buffer Overflow and Cross Scripting. The basic logic of PL/SQL injection is to attack the programs, which allows user's inputs. This input could be entry gate for entering hacker's own executable code. This method is used for example in passing through *DBMS\_ASSERT* (Oracle 10g R2) – that is used to verify the entered data. There is also another method called Dangling Cursors Snarfing. The principle is based on the fact, that Oracle doesn't close all cursors after it's usage. If privileged user would create cursor, it could be used by less privileged user to escalate rights on the more privileged user's level. To defend against this method the opened cursors should be closed right after it's usage. After escalating the privileges there is many things to do. One of it is to create Rootkit to create back door or to make any other malicious thing unseen.

Another method to escalate privileges is to decrypt passwords of other users from the *SYS.USER\$* table. Oracle is using hashing algorithm based on encryption algorithm DES. The principle of this encrypting algorithm is in using the password's salt. In Oracle, however there is weak salt choosing, character insensitivity and weak hashing algorithm. Access to the tables *SYS.USER\$* is bound to the access right *SELECT ANY DICTIONARY*. The attack vectors is to sniff the network communication, SQL injection or to access the *SYSTEM* table space (*system.dbf*) from the host operating system.

PL/SQL language is based on programming language ADA. PL/SQL allows to compile (wrap) the code into M-CODE, that is then passed to the Virtual Machine. In the 9i version there was possibility to guess the purpose of code thanks

**Listing 4.** Policy, that adds function *deny\_table\_rows* to the table *sec\_table*

```
BEGIN  DBMS_RLS.add_policy
(object_schema  => 'sec_user',
object_name    => 'sec_table',
policy_name    => 'sec_table_policy',
policy_function => 'deny_table_rows');
END;
```

**Listing 5.** Anonymous PL/SQL block that encrypts string in 256-bit AES

```
/* CRYPT IT ROUTINE IN AES 256-bit */
DECLARE
  k4y          RAW (32);
  t0p_s3cr3t_3nc  RAW (2000);
  t0p_s3cr3t_d3c  RAW (2000);
BEGIN
  /* 256 bit key - 32 byte */
  k4y := DBMS_CRYPTO.RANDOMBYTES(256/8);
  t0p_s3cr3t_3nc := DBMS_CRYPTO.ENCRYPT
  (
    src => UTL_I18N.STRING_TO_RAW ('h4x0rIzN0tD34d', 'AL32UTF8'),
    typ => 4360,
    /* encryption type - DBMS.CRYPTO.ENCRYPT_AES256 + DBMS.CRYPTO.CHAIN_CBC
       + DBMS.CRYPTO.PAD_PKCS5 */
    key => k4y
  );
  t0p_s3cr3t_d3c := DBMS_CRYPTO.DECRYPT
  (
    src => t0p_s3cr3t_3nc,
    typ => 4360,
    key => k4y
  );
  DBMS_OUTPUT.PUT_LINE (UTL_I18N.RAW_TO_CHAR (t0p_s3cr3t_d3c, 'AL32UTF8'));
END;
```

to reverse engineering. In that code there was visible the table of symbols (data structure, that points to the variable, function of data type in source code). In the version 10g the Symbol Table is not visible any more. Oracle 10g R2 has new feature to use wrapping by DBMS\_DLL (function CREATE\_WRAPPED).

Even for the Database System there could exist a worm. There is already Proof of Concept called Oracle Voyager Worm. This worm is trying to do some actions: grant DBA to PUBLIC, remove trigger and create trigger, that is run after database login and access Google, also it tries to send e-mail with the Oracle password Hashes. Then it tries to scan existence of another databases and it tried to connect by database link.

## Defending Oracle Database

The first task in securing the Database is physical restriction to the Database. It is must to secure the

database against user's physical access to protect the server from shut down or restart. The trend in implementation of authentication are biometric devices. These devices include fingerprinting, iris recognition or face recognition device.

Next step in the security is to protect Host operating system. This category include removing all unnecessary services (*ftp, telnet* etc.), enabling firewall and implementing security policies. Before plugging Oracle into the network it is must to control the access rights by each files and directories. Removing unnecessary user's accounts, removing unnecessary software and Intrusion Detection System (IDS) installing. One could remove banners to avoid operating system detection, running Anti-Virus, regular control the system, log monitoring and restrict number of super-users.

Except security of host operating system it is also important to secure the workstations. These could be

secured in different level according what purpose are these workstations used for (Database Administration, Development, Running Application). Some attack vectors could use features of SQL clients like TOAD or SQL\*Plus. The attack could be targeted on the files or records in the register, that could allow run some code after login. Many clients also store passwords. Even if the stored password is encrypted the encrypted password should be revealed.

In the field of Network security it's necessary to implement restriction of physical access to the network (e.g. limiting obtaining IP addresses with DHCP only for known MAC addresses). It is must to place Database Server behind the Firewall. Firewall must be placed out of the protected network that has to be protected and it's necessary to open only secured protocols and ports. Except this it is recommended to use Oracle Connection Manager. OCM could significantly help securing the network access to the Database Server. Also it is important to secure Oracle Listener, change default ports, use Node Filtering, that will filter clients on the IP Address base. One of common tasks should be Oracle Listener's Log checking.

There is an option in user's authentication. It is called Identification by Operating System. This option is no longer safe. It is not recommended to use it, because it's vulnerable. In the authentication process, it's good to define rights, roles, profiles and restrict available user's resources. Actual system rights could be obtained from the view USER\_SYS\_PRIVS. The access rights to the tables is stored in USER\_TAB\_PRIVS. The column ADMIN\_OPTION shows, if it is possible to grant rights to another user. Due to need of grouping the rights we can group it to the role. There are pre-defined roles – CONNECT, RESOURCE and DBA. It is must to take care, because e.g. the role CONNECT is not only for connecting user to the database, but it also allows to create tables, synonyms or views. To retrieve user's role

## On the Net

- [http://en.wikipedia.org/wiki/Oracle\\_Database](http://en.wikipedia.org/wiki/Oracle_Database),
- <http://www.oracle.com/database>,
- [http://www.red-database-security.com/whitepaper/oracle\\_default\\_ports.html](http://www.red-database-security.com/whitepaper/oracle_default_ports.html),
- <http://www.dokflead.net/duh/modules.php?name=News&file=article&sid=35>,
- <http://www.jammed.com/~jwa/hacks/security/tnscommand/tnscommand>,
- <http://www.ngssoftware.com/squirrelora.htm>,
- <http://xforce.iss.net/xforce/alerts/id/advise82>,
- <http://www.appsecinc.com/resources/alerts/oracle/02-0013.shtml>,
- <http://www.thc.org/thc-hydra/>,
- [http://www.cqure.net/wp/?page\\_id=3](http://www.cqure.net/wp/?page_id=3),
- <http://www.petefinnigan.com/orasec.htm>,
- [http://www.dba-oracle.com/t\\_oracle\\_biometrics\\_security.htm](http://www.dba-oracle.com/t_oracle_biometrics_security.htm),
- <http://www.databasejournal.com/features/oracle/article.php/3644956>.

## Reference

- Alexander Kornbrust, 2006. Oracle rootkits, Hakin9 1/2006,
- Joshua Wright, Carlos Sid, 2005. An Assessment of the Oracle Password Hashing Algorithm,
- Alexander Kornbrust, 2005. Hardening Oracle Administration – and Developer Workstations,
- William Heney, Marlene Theriault, 1998. O'Reilly – Oracle Security,
- David Know, 2004. Effective Oracle Database 10g Security,
- Integrity, 2004. Oracle Database Listener Security Guide,
- Pete Finningan, 2006. How to unwrap PL/SQL,
- Marlene Theriault, Aaron Newman, 2001. Oracle Security Handbook.



it should be used the view `USER_ROLE_PRIVS`. To protect Database resources the profiles could be used. Database records information about profiles in the view `DBA_PROFILES`. Administrator could create own profile (see Listing 1). In the profile it could be defined, how many retries has user to enter the password before the account will lock. `PASSWORD_LOCK_TIME` defines the time for how long will be the account locked after the maximum retries of entering password is reached. `PASSWORD_LIFE_TIME` defined the life time of the password in days. `PASSWORD_GRACE_TIME` defines the number of days before password expiration when Oracle displays the warning about the password expire. There is an interesting possibility is to create your own function (see Listing 2) that will check the password before it will be changed. The checking function could check the right length of the password or if it's not a dictionary word. The profile could be given to the user both in the time of user creating or additionally with the command:

```
ALTER USER n1c3_us3r PROFILE paranoid
```

Another security feature is to restrict the space in the tablespace. This could be done by command:

```
ALTER USER n1c3_us3r 100M ON USERS;
```

Further steps could be done to hack-proof the Oracle Database. One of this step could be installing only necessary components. It is recommended to use the principle of least possible configuration. The installed options could be retrieved from `V$OPTION` view. According to the attack vectors it is necessary to defend against the intruder, that checks default *usernames/passwords*. It could be good to lock these accounts by query `ALTER USER hr ACCOUNT LOCK` and/or change the password `ALTER USER hr IDENTIFIED BY n1c3n3wp4ss`; It is necessary not to give privileges type *ANY*. If this privilege is granted, there is pos-

## About the Author

Mikolas Pansky is employee of Czech computer company Cleverlance Enterprise Solutions as database developer. He is also PhD. student at the Charles University Faculty of Education, where he went after he has done his Master's degree in Informatics.

Contact with the author: [mikolas.pansky@gmail.com](mailto:mikolas.pansky@gmail.com)

sibility to work with Data Dictionary, which should be avoided. Extended protection of data dictionary could be done by adding initialization parameter `07_DICTIONARY_ACCESSIBILITY = FALSE`. This parameter will restrict the privilege `DELETE ANY`. It is good to give to the user only necessary privileges, nothing more. Another thing to do is to restrict default role `PUBLIC`. `PUBLIC` role is default for every new Oracle user. In the default configuration it allows working with some strong packages that could be compromised. These includes `UTL_SMTP` (for sending e-mails), `UTL_TCP` (for using TCP/IP), `UTL_HTTP` (Allows web access), `UTL_FILE` (for accessing the file system) and crypto package `DBMS_CCRYPTO`. Effective control could be reached by using initialization parameter `REMOTE_OS_AUTH = FALSE`. For common Administration tasks (start, shutdown, backup, recovery and archive) could be proffered the `SYSOPER` role (instead of `SYSDBA`).

Oracle Database offers the row-level security. This type of security is part of Virtual Private Database (VPD). VPD ensures basic security *rusel*. These defines PL/SQL function, that returns string. This function is then added to selected object (table, view or synonym), that we would like to protect with `DBMS_RLS` PL/SQL package. If then a SQL query is issued, Oracle adds to the end of query the returned string from defined function. This function then could be restriction, that removes rows, which contains in the column `user` value `SYS` (see Listing 3). The rule, that ensure, that the reply from the `SELECT` will not contain certain rows could be defined also by the `DBMS_RLS` package (see Listing 4). Further reading about this topic e.g. VPD article on

[www.databasejournal.com](http://www.databasejournal.com).

There are some reasons, why to encrypt the data in the database. One of it is for example the reason, that it is must to hide some information against DBA. Another is to reach some security standard. To encrypt the data it is possible to use `DMBS_CCRYPTO` package (it should replace `DBMS_OBFUSCATION_TOOLKIT`) in the future. `DBMS_CCRYPTO` is orientated on the work with data type `RAW`. That's not an obstruction due possibility to convert `VARCHAR2` to `RAW` and vice-versa with the package `UTL_RAW`. This package offers `DES` (not recommended any more), triple-`DES` with two `KEYS`, triple-`DES` with three keys, `AES` with various key length and algorithm `RC4`. The Listing 5 shows an example of 256-bit `AES` encrypting with Cipher-Block-Chaining according to the `PKCS#5` standard (see `RFC 2898`).

## Conclusion

I wanted this article to make an overview in basic security concepts of Database System Oracle from two different points of view: *attack and defense*.