Cryptanalysis of Password Protection of Oracle Database Management System (DBMS)

Timur Koishibayev¹, Zhanat Umarova²
¹Master student, South Kazakhstan State University, Information Systems department
Tauke Khan Avenue, 5, Shymkent, Kazakhstan
²Associated professor, PhD, South Kazakhstan State University,
Information Systems department Tauke Khan Avenue, 5, Shymkent, Kazakhstan
E-mail: koishibayevtimur@gmail.com, zhanat-u@mail.ru

Abstract. This article discusses the currently available encryption algorithms in the Oracle database, also the proposed upgraded encryption algorithm, which consists of 4 steps. In conclusion we make an analysis of password encryption of Oracle Database.

1. Introduction

One of the most characteristic features of the modern level of development of information technology is widespread computer networks. Computers, connected to the network, can significantly improve the effectiveness of the use of the computer system as a whole. Increase of productivity in such case is achieved through a network of information exchange, as well as by using on each computer general network resources [1]. The vulnerability of computer networks is connected with the support of different protocols of information exchange, with the length of communication channels. Therefore, on the first place we put the safety of the transferred information.

The classical scheme of databases’ (DB) protection is divided into the following obligatory procedures [2]:
- Access control - each user, including the administrator, has an access only to the necessary for him information according to his position;
- Access protection - the user can obtain an access to the data, who passed the last procedure of identification and authentication;
- Encryption - you must encrypt both data transmitted on the network to protect from eavesdropping and data written to the media, to protect from the theft and unauthorized viewing media / changes without funds of database (DBMS) management system, an access auditing to the database - activities with critical actions should be recorded.

Access to the protocol should not be used by the users on which it is conducted. In the case of applications, used a multi-tier architecture, the security features also take place, with the exception of data protection on the media - this function is left on the database [3].

With all listed security features in one way or another feature are equipped by database and with Oracle applications, which distinguish them from competitors' products. Oracle implements the SSL protocol to encrypt data that is transferred from the customer databases to the database and back. The advantage of SSL is that it is the current Internet standard, and can be used by clients who do not work with protocols of Oracle Net Services [4].

2. The Scheme of Encryption in Oracle Database

In the Oracle database to create a password can be used the following set of symbols [7]:
1. The figures '0123456789'
2. The letters 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
3. The symbols! "#$%& () + * - / : ; <= ? _ .


As well as all symbols are converted to uppercase, then the power of the alphabet passwords is $10 + 26 + 21 = 57$ symbols. As a result, the available passwords can be

$$\sum_{i=1}^{30} 57^i \approx 4.8 \cdot 10^{52}$$

(all passwords with length of 1 symbol, plus all passwords with length of two symbols and so on up to 30 symbols). However, in many cases this set a little smaller, because some symbols from item 3 are inconvenient to use, e.g., in a script shell.

Experiments have shown that as the password may be used any general symbols [5]

The key moment here is that the syntax of the Oracle server allows a construction with double quotation marks, among which can be any symbol.

On the database – the user’s passwords are stored as 8-byte hash in the schedule SYS.USER $.

In [5] this algorithm is described as follows:

1. Concatenate the username and password (this operation is denoted ||).
2. Convert the resulting string to uppercase (UPPER (Username || Password)).
3. If the operating system uses a single-byte encoding, then convert each double-byte symbol by filling the high byte of zeros (0x00).
4. Encrypt the resulting string (supplementing it with zeros to the length of the unit), using the DES (Data Encryption Standard) algorithm in CBC mode with a fixed key, the value of which has 0x0123456789ABCDEF.
5. Encrypt the resulting string again using the DES-CBC, but using the last block of the previous step as the encryption key.

Let $X$ encryption row on the key $K$ as a DES ($X, K$). Then, in an analytical form of encryption equation looks like this:

$$H = \text{DES} (\text{UPPER (Username||Password)}, \text{DES} (\text{UPPER (Username||Password)}, K))$$

Where $H$ - hash function.

If we denote $\text{UPPER (Username||Password)} = x$, then the equation will be quite simple:

$$H = \text{DES} (X, \text{DES} (X, K))$$

The inverse operation (decryption $H$ on the key $K$) we denote as:

$$X=\text{DES}^{-1}(H,K)$$

3. The Encryption Algorithm in the Oracle Database

Let the input of the algorithm receives an input data stream, the hash we should find [6]. Message length may be any (including zero). We write the message’s length in $L$. This number is an integer and non-negative. Multiplicity for any numbers is not required. After receipt of the data – there begin the process of flow’s preparation to calculations.

Step 1 Flattening of the stream.

First, we write till the end the single bit stream (bytes 0x80), then the required number of zero bits. Input data are aligned so that their new size $L'$ was comparable to 448 by module 512 ($L' = 512 \times N + 448$). Alignment occurs even if the length is comparable to 448.

Step 2 Adding the length of the message.

The remaining 64 bits appends a 64-bit representation of the data length (number of bits in the message) to the alignment. First we record lower 4 bytes. If the length exceeds 264 - 1, then appends only the least lower bits. After this the stream’s length will be a multiple of 512. The calculations will be based on the presentation of this data stream as an array of words of 512 bits.

Step 3 Initialize the buffer.

The variables with size from 4 to 32 bits are initialized for the calculations, and the initial values are set with hexadecimal numbers (the hexadecimal representation, low byte firstly):

$A = 01 23 45 67;$
$B = 89 AB CD EF;$
$C = FE DC BA 98;$
D = 76 54 32 10. 58
The results of intermediate calculations will be stored in these variables [7]. The initial state of
ABCD is called initialization vector.
Let define functions and constants, which we need for calculations.
It will take 4 functions (5) - (8) for four rounds. Let introduce a function of three parameters X, Y,
Z - the words; the result will also be a word.
Round 1
\[ \text{FunF}(X, Y, Z) = (X \land Y) \lor (\neg X \land Z) \] (5)
Round 2
\[ \text{FunG}(X, Y, Z) = (X \land Z) \lor (\neg Z \land Y) \] (6)
Round 3
\[ \text{FunH}(X, Y, Z) = X \oplus Y \oplus Z \] (7)
Round 4
\[ \text{FunI}(X, Y, Z) = Y \oplus (\neg Z \lor X) \] (8)
Let define constants array of data T [1...64], built as follows:
\[ T[i] = \text{int}(4294967296 \cdot \sin(i)) \] (9)
Where 4294967296 = \(2^{32}\)
The aligned data is divided into units (words) to 32 bits, and each block has 4 rounds of 16
operators. All operators are similar and have the form \([abcd k s i]\), defined by the formula (10):
\[ a = b + ((a + \text{Fun}(b, c, d) + X[k] + T[i]) \ll s) \] (10)
where X - a block of data;
\[ X[k] = M[n \cdot 16 + k] \] (11)
where k - number of 32-bit words from the n-th bit of the message block 512; s - the cyclic shift to the
left by s bit received from 32-bit argument.
Step 4 Calculating into cycle.
Let enter n element of the array in the data block. The values of A, B, C & D are stored remaining
after operations on the previous blocks (or their initial value if the block is the first).
AA = A
BB = B
CC = C
DD = D
Sum up with the result of the previous cycle: 59
A = AA + A
B = BB + B
C = CC + C
D = DD + D
After the end of the cycle, check whether there are more units for calculations. If so, change the
number of array elements (n ++) and go to the beginning of the cycle (figure1).
4. Analysis of the Reliability of Oracle Database

From an algorithmic point of view, during the opening of a modern cipher with a random key, there are two fundamentally different lines of attack: the research and the usage of some vulnerability of the algorithm, and brute force key [8].

Vulnerability - a long and painstaking analytic work, which allows drawing conclusions about the original data, based on the encrypted data and knowledge of the algorithm (cryptosystems).

Bust keys - (the second name "Power Attack", brute force - BF) – a standard method for selecting a password, which generate the next meaning, encrypting it and comparing the result with the existing hash. The main role in brute processing power play the computer powerful intruders, so it is a universal way to crack a variety of ciphers, since this method does not depend on the internal structure of the cipher.

In view of the large amount of time and computing resources, to the sorting we resort in the case, when other methods to solve the problem are impossible. And even in the case of brute force is still trying to narrow the set of sorted out the analysis of peculiarities of a particular algorithm [9].

In the future, we will assume that the attacker knows the user's login, H hash and the encryption key K.

If cryptosystem in Oracle was based on a single encryption, in which $H = DES \ (X, K)$, then the famous $H$ and $K$, it would instantly lose a resistance, as $X = DES^{-1} \ (H, K)$, and all the parameters of the right are known.

In Oracle, the situation is more difficult, and by our assumptions to calculate the password $X$, the equation would look like (4).
Note that for the reverse operation, it requires an intermediate key \( K \). So to compute the value of \( X \), in this equation we need the value \( K = \text{DES}(X, K) \), which is an intermediate key and is not stored anywhere. And calculate the DES \((X, K)\) is not possible because there is \( X||\text{Password Username} \) and the password for the conditions of the problem is unknown [10].

Written above means that, even knowing the encryption key \( K \), you can only calculate all possible values DES \((\text{Username}||\text{Password } K)\) for all possible values of a password, and then find in this set a suitable value. And it can be done by trying all possible passwords, i.e. in fact, only by the force of attack!

That is the main focus in the construction of the cryptographic Oracle: to calculate the secret password \( X \), you must first calculate the intermediate key DES \((\text{Username}||\text{Password } K)\) for all possible values of a password. In other words, the theoretical difficulty of breaking such cryptosystems with no real vulnerabilities, we use the complexity of power attack on this cipher.

In order to make sure in it perfect, you should explore its characteristics to identify laws that allow it to reduce the cryptographic strength [11]. On the view point of cryptography, used in Oracle a construction has two potentially interesting direction of analysis (attacks) [12].

1. The study of the properties of the intermediate key. Despite the fact that the DES is enough good algorithms, it can be expected that the subset generated values \( K ' \) will be small and will have special properties. As a result, from \( 2^{64} \) of the theoretically - possible values are \( K ' \), only a small part may in fact be realized. From the value of \( K ' \), together with the value of \( X \) can create a database to continue to \( K ' \) can be taken from the database and to input the second node DES.

2. Potentially interesting will be an attack with the accumulation DES \((X, K)\) on the one hand, and DES\(^{-1}\)(\(H,Y\)) on the other, wherein \( Y \) - a random value (the probable value of the intermediate key). In this case, it is possible so to parallelize this attack, as reduce its complexity.

5. Conclusion

After the above-mentioned changes - the crypto-scheme of Oracle Corporation should be subjected to certification. This will enhance the credibility of the software of Oracle. For such popular product like Oracle - an official recognition and confirmation of its reliability by the competent institutions’ sides is an important moment, both in its exploitation and in promoting the market. In the worst case, the weak link will be found and replaced by a more persistent [13].

As an option for building of a secure database, we could generate a unique key value for each database installation. The fact that for some IS with high security requirement [14]s, it is very desirable to be able to compile the own client with protected in it with a unique key, i.e. the client, who is able to connect only to his/her "native" database, equipped with the same key. It is clear that the ordinary customer with a standard key will never be connected to a database. Despite all the difficulties arise, the proposed construction provides a strong guarantee about the security of information in the database.

6. References

[1] Mikhalev V 2014 Comparative analysis of compliance of Oracle information security requirements of PCI DSS banking standards (Molodoy ucheniy) p 103-107
[2] Ochakov A and Orshanskiy A Information Protection by means of ORACLE database ways.
[3] Petrov A 2014 Controlling access to the MySQL and Oracle databases (Science Publishing Problems)
[4] Bondar A 2008 Data encryption in the Oracle database (Cryptographic aspects of information security)
[5] Pudovchenko Yu 2007 Encrypting passwords in the Oracle database
[6] Joshua W and Carlos C 2005 An Assessment of the Oracle Password Hashing Algorithm 2005
[7] Kulmugariev M 2014 Construction of information security systems for the software packages used in monopolistic access (master's thesis) (Almaty)
[8] Dodokhov A.L., Sabanov A.G, Investigation of data privacy protection on Oracle data base use, TUSUR reports, № 2 (24), part 2, December 2011.
[9] Oracle Database Security Guide. http://www.oracle.com/pls/db102/to_pdf?partno=b14266.
[10] Oracle Database Advanced Security Administrator’s Guide. http://www.oracle.com/pls/db102/to_pdf?partno=b14268.
[11] Dokhodov A and Sabanov A 2014 Investigation of the Oracle database for personal data protection / (Reports of the Tomsk State University of Control Systems and Radio Electronics) №2 (24) p 267-270.
[12] Dokhodov A 2014 On the question of the protection of personal data using Oracle database (web journal Fors)
[13] Ensuring the protection of personal data in the database Oracle, Protection of Personal Data, Data protection, and management of information security risks
[14] Pishyulin A 2013 Technology of Oracle Streams (The system administrator)