

[Skip to content](#)

[RSS](#)

[Email](#)

[Twitter](#)

[Sergio Hernando](#)

Seguridad de la Información, Análisis Forense y Auditoría de Sistemas

- [Home](#)
- [Buscador](#)
- [Contacto](#)
- [Empleo](#)
- [Legal](#)
- [Sobre el autor](#)

Tags

[Auditoria Oracle](#), [Contraseñas Oracle](#), [Oracle audit](#), [Oracle passwords](#), [Oracle security](#), [Seguridad Oracle](#)

Auditoría de contraseñas en Oracle Database (3 de 4): Fuerza bruta sobre claves Oracle

Publicado por Sergio Hernando el 2 marzo 2010

TEMARIO

[Auditoría de contraseñas en Oracle Database \(1 de 4\): Introducción y primeros pasos](#)

[Auditoría de contraseñas en Oracle Database \(2 de 4\): Adivinación de Oracle SID \(System ID\)](#)

[Auditoría de contraseñas en Oracle Database \(3 de 4\): Fuerza bruta sobre claves Oracle](#)

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continua navegando esta dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

plugin cooki

Buenas,

Continuando con lo explicado en los dos primeros artículos toca hablar de cómo utilizar ataques de fuerza bruta para evaluar la calidad de las claves Oracle. No obstante, antes de hacerlo, es preciso comprender dónde y cómo se almacenan las claves.

Distintas versiones, distintas ubicaciones

Las contraseñas Oracle se guardan, por norma general, en una tabla llamada `DBA_USERS`. Excepciones a esta norma hay muchas, así por ejemplo, `SYS.USER$`, o la tabla `USER$`, lugar donde se almacenan en la versión XE que estamos utilizando para esta serie de artículos.

```
SQL> select name,password from USER$ where password is not null;
```

NAME	PASSWORD
SYS	DCB748A5BC5390F2
SYSTEM	EED9B65CCECDB2E9
OUTLN	4A3BA55E08595C81
DIP	CE4A36B8ED6CA59C
TSMSYS	3DF26A8B17D0F29F
DBSNMP	ED66D214D5421CCC
CTXSYS	D1D21CA56994CAB6
XDB	E76A6BD999EF9FF1
ANONYMOUS	anonymous
MDSYS	72979A94BAD2AF8D
HR	4C6D73C3E8BDF0DA

NAME	PASSWORD
FLWS_FILES	364B78B9EABB9E56
FLWS_020100	16E4C012E98710DD
SHERNANDODBA	CB07C34C6F23F3D9
SHERNANDO	4CF6C046AD6673D7

Históricamente la presencia de numerosos usuarios de sistema en la configuración de fábrica ha supuesto y sigue suponiendo muchos problemas para la seguridad Oracle. Me recuerda al caso de los servidores de largo y medio alcance de IBM, donde los sistemas operativos suelen venir con una cantidad ingente de usuarios creados que si no son modificados pueden provocar un problema cuando el sistema entre en producción.

A lo largo de los años, y según han ido cambiando las versiones de Oracle, también han ido cambiando los métodos de generación y conservación de claves. Desde el cifrado de la concatenación de usuario y clave en las versiones entre 7 y 10g R2 hasta el empleo de *salts* en 11g, donde se emplea SHA-1 para generar un *hash* de la concatenación de clave y *salt*. Para cada versión el auditor debe repasar la documentación y comprender cómo y

http://www.red-database-security.com/whitepaper/oracle_passwords.html que puede servir de orientación.

Es importante reseñar que desde la versión 11g en adelante los *hashes* no se almacenan en `DBA_USERS`. En esta versión hay que recuperar los campos *name* y *spare4* de `SYS.USER$`.

```
SELECT name,spare4 FROM SYS.USER$ WHERE password is not null;
```

Tampoco entraremos a comentar otras ubicaciones donde puedan estar presentes los *hashes* de las claves, como las vistas que se hayan creado, duplicados de las tablas, *rollbacks*, copias de seguridad, etc. En definitiva, la misión del auditor es identificar dónde están las claves y obtener la relación usuario-*hash* para poder así realizar una verificación de calidad de las mismas. También es importante determinar quién puede obtener esos datos, qué permisos son necesarios y por supuesto, hay que indagar en todos los métodos posibles para obtenerlos, incluyendo el *pentesting*.

Caja negra, gris y blanca

En el primero de los casos, las pruebas irán encaminadas, mediante *pentesting*, a hacernos con un listado de usuarios y *hash* de clave para tratar de obtener claves en claro. El ataque constará de dos partes:

- Intrusión al sistema, habitualmente por cuentas con clave por defecto no modificadas o bien aprovechando algún *exploit* que nos permita el acceso.
- Extracción de relación de usuarios y *hashes*, bien por haber logrado acceso privilegiado que permita hacerlo directamente, bien por provocar escalada de privilegios que nos permita finalmente obtener la relación. En algunos casos, si la configuración de seguridad no es adecuada, bastará con tener los privilegios mínimos para obtener la relación.

En el caso de caja gris el auditor tratará de obtener la relación de usuarios empleando las credenciales no privilegiadas que le han sido entregadas. Es frecuente, por mala configuración, que incluso los perfiles más bajos puedan obtener la relación de usuarios y *hashes*. Este modelo es análogo al de caja negra, pero con la diferencia de que el auditor contará de partida con una cuenta en el sistema.

Los usuarios especiales y los datos críticos, el botón más cotizado

Siempre que sea posible, habida cuenta de que por defecto son las cuentas más poderosas, el análisis tendrá en cuenta especialmente los siguientes usuarios: SYS, SYSTEM, SYSMAN y DBSNMP. En la documentación hallaréis descripciones de cada uno de estos usuarios.

No debemos perder de vista los usuarios con privilegios elevados (DBA y otros) que se hayan podido crear tras la instalación. Basta con descuidar un sólo perfil privilegiado para que un atacante tome control total de la instalación. Es tan importante que SYS disponga de una clave de alta calidad como cualquier otro usuario creado con perfil suficiente para acceder a los datos críticos. No olvidemos que en Oracle la seguridad debe siempre enfocarse a la seguridad de los datos, así pues, si el usuario SHERNANDO tiene privilegios mínimos globales pero es dueño de la tabla más importante del sistema, su compromiso es equivalente al de comprometer la cuenta SYS.

Fuerza bruta sobre las claves

En nuestro ejemplo vamos a crear algunos usuarios adicionales con distintas claves:

```
SQL> create user shernando2 identified by pass2;
Usuario creado.
SQL> create user shernando3 identified by pass3;
Usuario creado.
SQL> create user shernando4 identified by pass4;
Usuario creado.
SQL> create user shernando5 identified by pass5;
Usuario creado.
```

A continuación, sacaremos a fichero todas las relaciones usuario-hash, empleando el comando `spool`.

```
SQL> spool usuarios.txt
SQL> select name,password from USER$ where password is not null;

NAME                                PASSWORD
-----
SYS                                DCB748A5BC5390F2
SYSTEM                            EED9B65CCECDB2E9
OUTLN                             4A3BA55E08595C81
DIP                               CE4A36B8E06CA59C
TSMSYS                           3DF26A8B17D0F29F
SHERNAND05                       142A83958AD6AEB1
DBSNMP                           E066D214D5421CCC
CTXSYS                           D1D21CA56994CAB6
XDB                              E76A6BD999EF9FF1
ANONYMOUS                        anonymous
MDSYS                            72979A94BAD2AF80

NAME                                PASSWORD
-----
HR                                4C6D73C3E8B0F0DA
FLOWS_FILES                      364B78B9EABB9E56
FLOWS_020100                    16E4C012E98710D0
SHERNAND0DBA                    CB07C34C6F23F3D9
SHERNAND0                       4CF6C046AD667307
SHERNAND02                      62A93F338C682143
SHERNAND03                      9600696A889C8B6C
SHERNAND04                      968B16ECB8D7924C

19 filas seleccionadas.

SQL> spool off
```

Recogemos el fichero, en este caso de `C:\XEClient\bin` y lo ponemos a buen recaudo. He dejado una copia del fichero en http://www.sahw.com/images/oracle_audit/usuarios.txt por si queréis utilizarlo.

Hay infinidad de programas para someter a los hashes obtenidos a fuerza bruta. [Uno de los más populares es orabf](#), que incluye una lista de claves habituales y diversas opciones para construir los ataques de fuerza bruta. Veamos un ejemplo de cómo se utiliza:

```
C:\tools\orabf-v0.7.6>orabf

orabf v0.7.6, (C)2005 orm@toolcrypt.org
-----

usage: orabf [hash]:[username] [options]

options:
-c [num]  complexity: a number in [1..6] or a filename
-         read words from stdin
[file]   read words from file
1        numbers
2        alpha
3        alphanum
4        standard oracle (alpha)(alpha,num,_,#,$)... (default)
5        entire keyspace (' '..~')
6        custom (charset read from first line of file: charset.orabf)
-m [num]  max pwd len: must be in the interval [1..14] (default: 14)
-n [num]  min pwd len: must be in the interval [1..14] (default: 1)
-r        resume: tries to resume a previous session

C:\tools\orabf-v0.7.6>orabf DCB748A5BC5390F2:SYS

orabf v0.7.6, (C)2005 orm@toolcrypt.org
```

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continua navegando esta dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

plugin cooki

En el ejemplo anterior vemos cómo el usuario SYS tiene una clave por defecto llamada password, que ni siquiera ha sido preciso romper mediante fuerza bruta puesto que estaba en el diccionario. Otro ejemplo:

```
C:\tools\orabf-v0.7.6>orabf 62A93F338C682143:SHERNANDO2
orabf v0.7.6, (C)2005 orm@toolcrypt.org
-----
Trying default passwords...done

Starting brute force session using charset:
#$0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ_
press 'q' to quit. any other key to see status
password found: SHERNANDO2:PASS2
37043128 passwords tried. elapsed time 00:01:04. t/s:577952
```

En este caso ha bastado un minuto para averiguar la clave. Veamos qué pasa si cambiamos la clave de pass2 a clave2 (ampliamos de 5 a 6 caracteres)

```
SQL> alter user shernando2 identified by clave2;
Usuario modificado.

SQL> select name,password from USER$ where name like 'SHERNANDO2';

NAME                                PASSWORD
-----
SHERNANDO2                          2E14487E283D2E62
```

Sometemos el nuevo hash a fuerza bruta:

```
C:\tools\orabf-v0.7.6>orabf 2E14487E283D2E62:SHERNANDO2
orabf v0.7.6, (C)2005 orm@toolcrypt.org
-----
Trying default passwords...done

Starting brute force session using charset:
#$0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ_
press 'q' to quit. any other key to see status
password found: SHERNANDO2:CLAVE2
296152537 passwords tried. elapsed time 00:08:31. t/s:579059
```

Como era previsible, el tiempo de ataque se ha elevado a varios minutos. A mayor calidad de la clave, más tiempo. Por lo general el tiempo de cómputo elevado no hace rentable los intentos de fuerza bruta en claves complejas de 7 o más caracteres.

Existen infinidad de herramientas para fuerza bruta: herramientas W32 como la maestrada, LINUX procedimientos PL/SQL scripts Perl, etc. También una buena selección de

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando esta dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

plugin cooki

Fuerza bruta con múltiples cuentas

Entre las muchas herramientas existentes [siempre recomiendo Inguma](#), de Joxean Koret, un *framework* de seguridad bastante completo que además tiene funcionalidades específicas para Oracle. Además de su versatilidad, Inguma está escrita en Python, con lo que es multiplataforma, y para el caso que nos ocupa tiene un módulo de fuerza bruta para contraseñas Oracle llamado *bruteora* que podemos lanzar contra las cuentas usuales en la base de datos:

```
inguma> sid ="%E"
inguma> exploit
Target: 127.0.0.1
Select module [all]: bruteora
[+] No user specified, trying ALL posible users
Brute forcing started
-----
Trying SYS/password.....
[+] Guessed SYS/password
```

Al ser código abierto, es fácil incorporar las cuentas declaradas en la base de datos para complementar las cuentas frecuentes empleadas por Inguma, con lo que finalmente es posible obtener también una valoración de la calidad de todas las contraseñas que existan en el sistema. Este método, a diferencia de otros, es *online*, con lo que el auditor debe tener presente el impacto sobre el sistema a auditar.

Un saludo,

Be Sociable, Share!

+ MORE

Tweet

Me gusta

G+

Share

Categoría/s → [Auditoria](#), [Seguridad](#)

4 comentarios →

1.



5 marzo 2010


luna [permalink](#)

Quiero contactarme via mail con vos para hacerte algunas preguntas por favor

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continua navegando esta dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

plugin cooki


2.  6 marzo 2010

[sergio hernando](#) [permalink](#)

@luna,

Usa el formulario de contacto, y si la duda es sobre Oracle, dejala aqui que la veamos todos, asi la podemos comentar mas personas.

Un saludo

3.  17 marzo 2010

Augusto [permalink](#)

Sergio excelente articulo y de mucha ayuda en nuestra labor de contribuir al aseguramiento de los sistemas de información. Me gustaria saber si tienes algun articulo sobre aseguramiento de bases de datos en sql server. Listado de usuarios genericos y hash respectivos y herramientas que se puedan utilizar para auditarla. gracias

Trackbacks & Pingbacks

1. [Auditoría de contraseñas en Oracle Database \(3 de 4\). Fuerza bruta sobre claves Oracle / DbRunas](#)

Escribir un comentario

Nombre: (required):	<input type="text"/>
Email: (required):	<input type="text"/>
Web:	<input type="text"/>

Comentario:

Note: XHTML permitido. Tu email nunca será publicado.

[Suscribirse a los comentarios via RSS](#)

Enviar comentario

Hospedaje cortesía de



• Recent Comments

- [inflatable paddle board packages](#): *Respect to author, some excellent selective information.*
- [private holiday lets in Whitby](#): *I'm very happy to uncover this website. I want too to thank you for your time...*
- [Dianne](#): *The game itself is actually a much more about interpersonal relationship occasionally as opposed to cards...*
- [parts inside](#): *Thank you, I've recently been looking for info approximately this topic for a long time and yours...*
- [send flowers to ahmedabad india](#): *Hurrah! In the end I ggot a weblog from where I be able to actually take valuable...*

• 10 últimos artículos

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continua navegando esta dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

plugin cooki

- [Instalar y usar LaTeX en Windows \(Actualización 2012\)](#)
- [España: Suicidio científico ... y tecnológico](#)
- [Snort en un router DD-WRT: Instalación, configuración y operación básica en 10 pasos](#)
- [Auditoría de centros de procesamiento de datos: Parte 3: Aspectos contractuales y de gestión energética](#)
- [Dando un nuevo paso](#)
- [Auditoría de centros de procesamiento de datos: Parte 2: Seguridad lógica](#)
- [Ocho](#)
- [Ha llegado ese momento del año en el que ...](#)
- [Auditoría de centros de procesamiento de datos: Parte 1: Seguridad física](#)

• Etiquetas

[Análisis Forense](#) [Antivirus](#) [AS/400](#) [AS/400 Security](#) [Audit](#) [Auditoria](#) [Cloud computing](#)

[crimeware](#) [Crisis](#) [EMV](#) [F-Secure](#) [financial malware](#) [Firefox](#) [Forensics](#) [Fraud](#) [Fraude](#) [Google](#)

[Humor](#) [i5/OS](#) [IBM](#) [IBM i](#) [IBM System i](#) [INTECO](#) [Internet Explorer](#) [IT](#) [Linux](#) [Mainframe](#) [Malware](#)

[Microsoft](#) [Mozilla](#) [Nube](#) [Online banking](#) [OS/400](#) [Phishing](#) [Privacidad](#) [Proteccion de datos](#) [Security](#)

[Seguridad](#) [Seguridad AS/400](#) [Spam](#) [Tarjetas](#) [Trojan](#) [Vulnerabilidades](#) [Vulnerabilities](#)

[Windows](#)

Licencia



Este blog está licenciado bajo [Creative Commons](#).

• Archivos

Archivos

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continua navegando esta dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

plugin cooki