Certificación y controles de seguridad en Bases de Datos: Introducción

Miguel Expósito Martín

Universidad de Cantabria

miguel.exposito@unican.es

26/11/2018

Visión general

Introducción

Contexto

Contexto

Hoy en día, las bases de datos dan soporte al almacenamiento de todo tipo de información sensible y crítica: cuentas bancarias, balances, datos fiscales, informes médicos, registros sobre empleo, compras con tarjetas de crédito, expedientes académicos...

Sin embargo, la seguridad en bases de datos ha sido muchas veces un aspecto al que **no se le ha dedicado la atención suficiente**. Con la aparición de normas como el Esquema Nacional de Seguridad o la ISO 27001 y la necesidad cada vez mayor de responder ante clientes y usuarios el escenario está cambiando y se está volviendo a poner el foco en la relevancia de estas cuestiones.

Contexto

La mayor parte de las veces, la seguridad en una base de datos consiste en aplicar los **principios ya probados** y utilizados en el ámbito de las redes de comunicaciones durante décadas:

- Filosofía del mínimo privilegio.
- Eliminar funcionalidad innecesaria.
- Ser estricto en la autenticación y el control de acceso.
- Utilizar encriptación cuando sea necesario.

¿Qué base de datos es la más segura?

Según datos de la lista de *Common Vulnerabilities and Exposures*:

BD	**# Bugs**
Oracle	430
MySQL	259
Postgresql	106
SQL Server	83
MongoDB	13

Cuadro: Vulnerabilidades registradas en CVE[1]

[1] http://cve.mitre.org/

¿Qué base de datos es la más segura?

Estos datos deben interpretarse **con cautela**: los inicios de las series históricas varían según el proveedor y no todas las vulnerabilidades existentes tienen por qué estar registradas. Además, no todas las bases de datos tienen el mismo nivel de escrutinio por parte de investigadores o posibles atacantes. Incluso la definición del producto "base de datos" puede ser problemática: Oracle y Microsoft suministran otros componentes junto con sus bases de datos.

Incluso aunque fuera posible obtener una métrica combinada, sólo se estarían considerando aspectos "parcheables" y no las características inherentes de seguridad que proporciona cada base de datos.

Opinión

Determinados autores[Litchfield, 2005] consideran que PostgreSQL es posiblemente la base de datos más enfocada a la seguridad por defecto.

Normativa aplicable

Normativa aplicable

- Reglamento General de Protección de Datos (RGPD)
- UNE-EN ISO/IEC 27001
- En la Administración Pública, Esquema Nacional de Seguridad (ENS)

RGPD

El Reglamento General de Protección de Datos (RGPD), de **obligado cumplimiento tanto en el ámbito público como en el privado desde el 25 de mayo de 2018** (siempre y cuando se recopilen y hagan tratamientos de datos de personas físicas), refuerza y prevalece sobre ciertos aspectos de la LOPD y el RD que la desarrolla. Incorpora novedades que lo hacen más restrictivo que la antigua normativa, tanto en materia de obligaciones como en sanciones.

Novedades por Wolters Kluwer
Tres ideas sobre sus implicaciones, por Borja Adsuara
Guía para responsables de tratamiento (AEPD)
Texto original en el BOE

RGPD: guía práctica de adaptación en la empresa privada

Hoja de ruta para adaptación al RGPD:

- Designar un DPD
- Elaborar el registro de actividades de tratamiento
- Realizar un análisis de riesgos
- Revisar medidas de seguridad a la luz del AR
- Establecer mecanismos y procedimientos de notificación de quiebras de seguridad
- A partir del AR, realizar, en su caso, una evaluación de impacto en la protección de datos

RGPD: guía práctica de adaptación en la empresa privada

Acciones simultáneas a los pasos anteriores:

- Adecuar formularios (derecho de información)
- Adaptar mecanismos y procedimientos para el ejercicio de derechos
- Adaptar los contratos y valorar si los encargados ofrecen garantías
- Elaborar o adaptar la política de privacidad

RGPD: medidas a aplicar

¿Qué medidas técnicas aplicar?

En el RGPD, los responsables y encargados establecerán las medidas técnicas y organizativas apropiadas para garantizar un nivel de seguridad adecuado en función de los riesgos detectados en el análisis previo.

Como referencia:

- Las propuestas en el *Esquema Nacional de Seguridad*
- Las propuestas en el *Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal*

RGPD: medidas a aplicar

Ejemplos de medidas[2] **directamente relacionadas** con bases de datos:

- Proceso de autorización
- Dimensionamiento / gestión de capacidades
- Protección de las comunicaciones
- Protección de la información:
  - Calificación de la información.
  - Cifrado de la información.
  - Copias de seguridad.
- Protección de servicios y aplicaciones web: inyección de código.
- Control de acceso.
- Identificación y autenticación.
- Auditoría.
- Cifrado en las telecomunicaciones.

[2] Fuente: ENS

Dimensiones de la seguridad TI

Dimensiones de la seguridad TI

Hay tres dimensiones clave en la seguridad de cualquier sistema de información, y por lo tanto, de un Sistema de Gestión de Bases de Datos (SGBD):

- Disponibilidad
- Integridad
- Confidencialidad

A estas dimensiones básicas se les suelen añadir otras dos:

- Autenticidad
- Trazabilidad

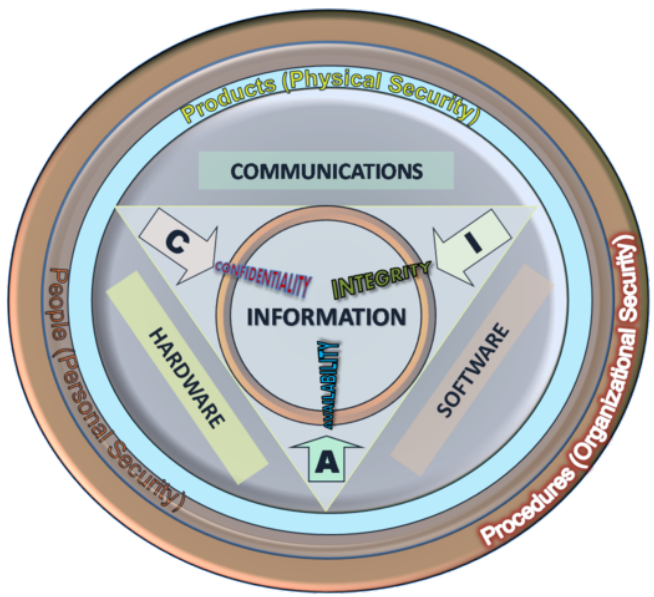

Figura: Triángulo CIA. Wikipedia.

Disponibilidad

- Los datos tienen que estar disponibles el tiempo que sea necesario.
- Los datos tienen que estar disponibles tan sólo para los usuarios apropiados.

Ejemplos de no disponibilidad

- Imposibilidad de acceder al correo electrónico por un error de configuración.
- Caída de un sistema debido a un DoS.

Integridad

- Es necesario verificar que cualquier dato de entrada es preciso y verificable, correcto y libre de modificaciones y/o errores.
- Deben existir mecanismos que eviten que los datos sean adulterados o corrompidos.

Ejemplos de ataques contra la integridad

- Alteración malintencionada de los ficheros de un sistema de información explotando una vulnerabilidad.
- Modificación de un informe de ventas por un empleado malintencionado o por un error humano.

Confidencialidad

- Hay que asegurar que los datos confidenciales solo están a disposición de determinados usuarios autorizados.
- Deben existir mecanismos que impidan la extracción y revelación no autorizada de datos.
- Han de asegurarse los datos frente a posibles brechas de seguridad internas o externas.

Ejemplos de falta de confidencialidad

- Robo de información por parte de un atacante a través de Internet.
- Divulgación no autorizada a través de redes.
- Acceso por parte de un empleado a información crítica de una Organización a la que no debería tener acceso.

Autenticidad

- Es necesario asegurar que los datos se modifican por una fuente autorizada.
- Es necesario confirmar que los usuarios que acceden al sistema son quienes dicen ser.
- Es necesario verificar que cualquier dato de salida está destinado al receptor esperado.

Ejemplos de falta de autenticidad

- Firma de un documento por otra persona haciéndose pasar por el firmante.
- Uso de un usuario y una contraseña de un compañero de trabajo.

Trazabilidad

- Debe de ser posible auditar quién ha accedido a qué datos.
- Debe existir un mecanismo que informe de cualquier cambio en los datos así como su autoría, para garantizar el cumplimiento de las reglas de negocio y la normativa de protección de datos.
- Es necesario informar de qué se ha hecho con ellos.

Ejemplos de falta de trazabilidad

- No saber qué operador ha hecho la última operación de mantenimiento en una base de datos.
- No saber qué usuario ha borrado una tabla de una base de datos en un momento dado.

Ejercicio 1.1 (1/2)

El CCN ofrece una guía para valorar sistemas de información y categorizarlas según el ENS. La guía presenta tablas-resumen que ayudan en la categorización; dichas tablas se estructuran de la siguiente forma:

- Criterios comunes a todas las dimensiones de tipos de información y servicios
  - Disposición legal
  - Perjuicio directo al ciudadano
  - Incumplimiento de norma
  - Pérdidas económicas
  - Reputación
  - Protestas
  - Delitos
- Criterios para tipos de información con datos personales en función del tipo.
- Criterios para tipos de información con datos personales en función del tratamiento.
- Criterios para la disponibilidad de servicios.

Ejercicio 1.1 (2/2)

En base a esta guía, categorice, **justificadamente**, las siguientes bases de datos, entendidas en su definición más genérica:

1. Sistema de Información Contable de una Comunidad Autónoma
2. Sistema de gestión de multas de la DGT
3. Banco de datos del Instituto Nacional de Estadística
4. BD documental de informes de auditoría de control financiero
5. Historial médico de pacientes del SCS
6. Sede electrónica del punto de acceso general de la AGE
7. Portal de datos abiertos del Ayuntamiento de Santander

Atención

Basta con que una de las dimensiones se categorice como nivel alto o medio para que todo el sistema de información sea categorizado de la misma forma. Intente encontrar un criterio por el cual el sistema se categorice como alto o medio.

Principales problemas de seguridad en BBDD

Principales amenazas en la seguridad de TI

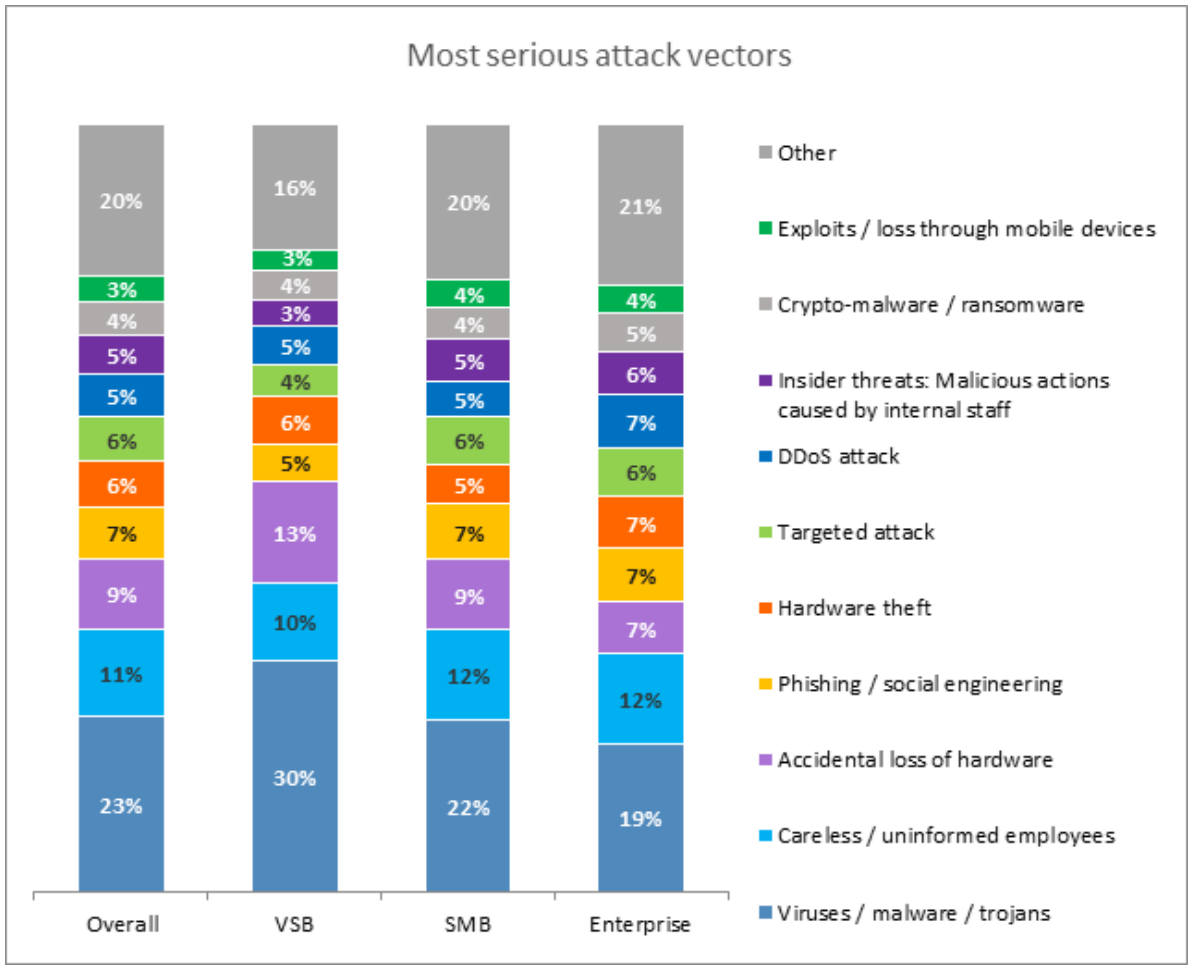

Figura: Principales vectores de ataque.[3]

[3] Fuente: [Kaspersky, 2017]

Principales amenazas en la seguridad de bases de datos

Es posible identificar las siguientes amenazas comunes en seguridad de bases de datos:

- Mal diseño de privilegios de acceso.
- Malware.
- Poca experiencia y formación en seguridad de bases de datos.
- Inyección SQL.
- Exploits de bases de datos vulnerables.

Mal diseño de privilegios de acceso

Muchas veces, por error o debido a políticas de acceso pobres, los privilegios otorgados a un usuario sobre la base de datos **exceden los requisitos de su puesto de trabajo**. También pueden permanecer sin modificar en el caso de que un trabajador cambie de rol o sea baja en la organización. Para evitar este tipo de situaciones, se dan dos recomendaciones:

- Diseñar una política de concesión de privilegios de acceso ajustada a los roles necesarios en la organización.
- Realizar **auditorías periódicas** en las que se comprueben roles y reglas de acceso, llevando a cabo los cambios necesarios para evitar cualquier exposición no deseada.

Atención: aviso a los usuarios

Si se hacen cambios en los privilegios de acceso, los usuarios lo notarán. Una buena política de comunicación explicando el por qué de los ajustes siempre ayudará a evitar más problemas de la cuenta.

Malware

Infecciones comunes a través del correo electrónico, usando técnicas como el *spear phising*, pueden convertir a los usuarios de bases de datos en conductos de entrada para que los atacantes ganen acceso a redes y datos sensibles. Para incrementar la protección sobre este tipo de ataques, de difícil detección, se recomiendan las siguientes acciones preventivas:

- Campañas de **comunicación, concienciación y formación** a los usuarios del correo, para que sean capaces de identificar este tipo de amenazas.
- Últimos avances en tecnologías de seguridad para el correo electrónico (filtros, inspección de URLs...)

Atención: el correo electrónico puede llegar a ser inutilizable

Si los filtros se aplican sin control o los sistemas de seguridad no están correctamente dimensionados en función del número de usuarios de la red, pueden darse problemas irritantes: eliminación de archivos adjuntos, clasificación de remitentes lícitos en SPAM, reescritura de URLs o datos...

Poca experiencia y formación en seguridad de bbdd

Muchos de los incidentes de seguridad en TI están relacionados con el *"factor humano"*. Estos incidentes podrían prevenirse siguiendo buenas prácticas en la implementación de controles internos y a través de campañas de formación, comunicación y concienciación de usuarios.

Las tareas que ayudan a mejorar la seguridad de TI están muy relacionadas con la calidad; el personal interno debe estar al día y adecuadamente formado en tecnologías de bases de datos y seguridad. Los órganos directivos de las organizaciones deben comprender la **importancia de dedicar recursos a este tipo de tareas**, no siendo aceptable un argumento de reducción de costes para no hacerlo.

Recordatorio

La calidad es gratis - Crosby

Inyección SQL

Cualquier componente que cree y ejecute consultas SQL dinámicamente está sujeto, en teoría, a ataques de inyección SQL. Este tipo de ataque permite a un atacante inyectar código para ejecutar comandos remotos que puedan leer o modificar una base de datos, o ser utilizados para escalar en privilegios. Es uno de los ataques a bases de datos más comunes y con el que están relacionadas algunas de las vulnerabilidades más antiguas de distintos BD.

Atención: causas más comunes

- Validación de entrada de usuario insuficiente en aplicaciones web.
- Construcción incorrecta de sentencias SQL en aplicaciones web.

Inyección SQL

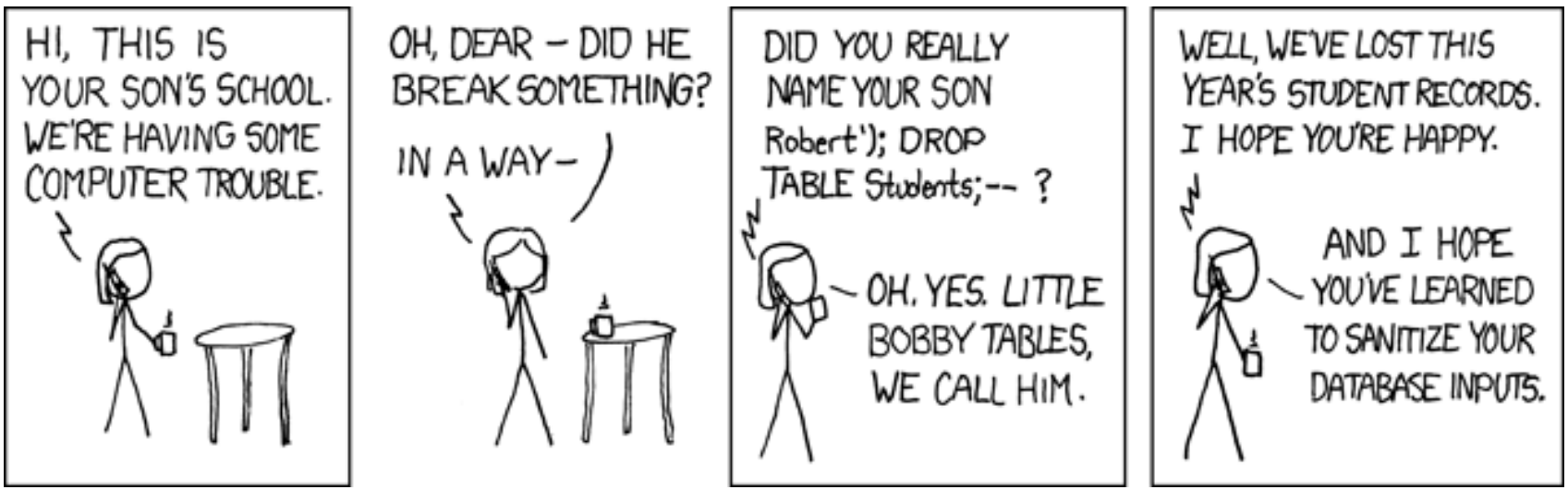

Figura: Little Bobby Tables.[4]

[4] Fuente: https://xkcd.com/327/

Inyección SQL: ejemplo trivial 1/2

Formulario web

```
<form action="/cgi-bin/login" method=post>
Username: <input type=text name=username>
Password: <input type=password name=password>
<input type=submit value=Login>
```

Enviado en la request

```
username=submittedUser&password=submittedPassword
```

Inyección SQL: ejemplo trivial 2/2

Posible SQL en la aplicación

```
select * from Users where (username = 'submittedUser'
and password = 'submittedPassword');
```

Envío del atacante

```
username=admin%27%29+--+&password=+
```

Consulta resultante

```
select * from Users where (username = 'admin') --
and password = ' ');
```

Simulación de inyección SQL

¿Cómo evitar este tipo de ataques?

- Validando la entrada del usuario.
- No utilizando sentencias SQL construidas dinámicamente.
- No utilizando cuentas con privilegios de administrador.
- No proporcionando más información de la estrictamente necesaria.

En el caso presentado, escapando los caracteres especiales en los formularios. Ejemplo: "debería transformarse en \".

Inyección SQL

Ejemplo en Java con JDBC

```
Connection con = (acquire Connection)
PreparedStatement query = con.prepareStatement(
"SELECT * FROM Usuarios WHERE user=? AND pass=?");
query.setString(1, user);
query.setString(2, password);
ResultSet rset = query.executeQuery();
```

Utilizando *prepared statements*, la entrada del usuario se trata como el contenido de un parámetro y no como parte de la propia sentencia SQL. En el caso de *Bobby Tables*, con el ejemplo anterior se insertaría el texto *"Robert'); DROP TABLE students; –"* en la columna del nombre del alumno.

Ejercicio 1.2

Dado el siguiente código para un entorno Python/MySQL:

Ejecución de SQL desde Python

```
cmd = "update people set name='{0}'
       where id='{1}'".format(name, id)
cursor.execute(cmd)
```

Indique si está afectado por la vulnerabilidad de inyección SQL y, de ser así, proponga una alternativa segura.

Ayuda

Documentación sobre la clase MySQLCursor
Documentación sobre el método format

Exploits en bases de datos vulnerables

La mayor parte de fallos explotables de
seguridad en bases de datos pueden clasificarse en las siguientes categorías:

- Fallos de autenticación en protocolos de red.
- Fallos en los propios protocolos de autenticación.
- Ejecución arbitraria de código en elementos SQL.
- Escalado de privilegios locales.

Exploits: fallos de autenticación en protocolos de red

Uno de los problemas más famosos en esta categoría es el gusano *"Slammer"* (75.000 víctimas en diez minutos), para SQL Server. El servicio de resolución de SQL Server opera en UDP 1434, exponiendo una serie de funciones; dos de ellas eran vulnerables a ataques de buffer overflow. No era necesaria autenticación para explotar estas vulnerabilidades.

Las mejores formas de protegerse de este tipo de ataques son:

- Parchear los SGBD al día.
- Asegurarse de que sólo pueden conectarse a los SGBD hosts de confianza.
- Implementar algún tipo de IPS/IDS.

Fallos autenticados en protocolos de red

Hay bastante menos bugs en esta categoría. La mejor protección contra este tipo de fallos es disponer de una política de contraseñas robusta.

Exploits: fallos en los protocolos de autenticación

Históricamente, muchos SGBD utilizaban protocolos de autenticación en texto plano (o con algún tipo de encriptación débil). Por ejemplo, la base matemática del algoritmo de autenticación de *MySQL* $<$ *4.1* fue puesta en cuestión por la capacidad de un atacante de determinar el *hash* de una contraseña simplemente observando varias autenticaciones.

Como medidas para evitar este tipo de ataques, se tienen las siguientes:

- Parchear los SGBD al día.
- Deshabilitar cualquier posible sistema de autenticación en texto plano.

Exploits: ejecución arbitraria de código en elementos SQL

Son ataques de tipo *buffer overflow* aplicados a elementos de la gramática de SQL que no están sujetos a los controles de acceso tradicionales (GRANT/REVOKE). Este tipo de bugs suelen ser explotados a través de problemas de inyección SQL.

Ejecución arbitraria en Oracle

Oracle presentó un bug que permitía a un atacante ejecutar código arbitrario a través de las funciones NYMTOYMINTERVAL, NUMTODSINTERVAL y FROM_TZ o el parámetro de sesión TIME_ZONE.

La mejor defensa contra este tipo de ataques es parchear los SGBD al día.

Inciso: ¿qué es un buffer overflow?

Una situación donde un programa en ejecución intenta escribir datos en posiciones fuera del buffer de memoria asignado para ello, en las que no debería escribir.

Buffer overflow

Introducción de un nombre de usuario de 10 caracteres cuando el tamaño de buffer es de 8. Si no se valida la entrada, se produce un desbordamiento.

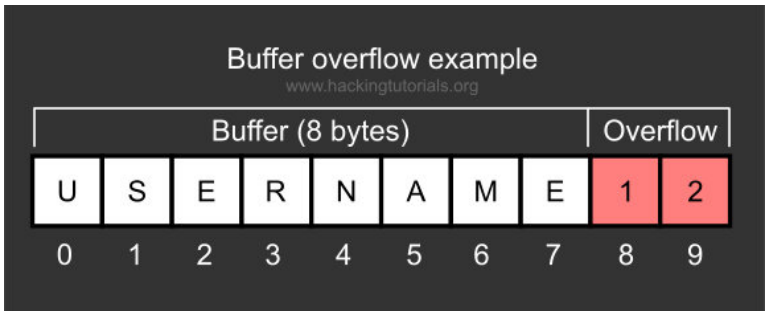

Figura: Buffer overflow. Fuente

Un atacante puede llegar a escribir código malicioso en esas posiciones de memoria del buffer desbordado y conseguir escalado de privilegios si el programa se ejecuta con privilegios de sistema.

Exploits: Escalado de privilegios locales.

Esta categoría se compone de bugs que permiten algún tipo de escalado de privilegios a nivel de sistema operativo.

Escalado de privilegios locales

- Oracle *"extproc"*.
- Procedimientos almacenados extendidos de SQL Server y Sybase.

La mejor defensa contra este tipo de ataques es **ejecutar la base de datos como un usuario con pocos privilegios**, o en una parte segregada del sistema de archivos en la cual solo la base de datos pueda realizar operaciones de lectura/escritura.

Resumen

- Si no se detectan más vulnerabilidades, es porque no se buscan.
- La normativa vigente obliga a aplicar medidas de seguridad también a las BBDD.
- La inyección SQL es el ataque “estrella”.
- No existe una base de datos “más segura”.
- La mejor prevención: mantener los sistemas actualizados y parcheados y aplicar seguridad por defecto.

Ejercicio 1.3

Una unidad de una organización pública cuyo negocio fundamental es el tratamiento de datos debe poner en marcha un SGBD en el que se almacenarán datos sobre solicitantes de subvenciones. Un problema en su disponibilidad reduciría significativamente la capacidad de la organización para atender la recogida de datos de los solicitantes, aunque podrían seguir desempeñando su trabajo (por ejemplo, tomando nota de los datos a mano.)

Seleccione **motivadamente** las medidas del Esquema Nacional de Seguridad **directamente relacionadas con SGBDs** que habría que aplicar al SGBD.

Criterios de evaluación: identificación correcta de medidas, justificaciones claras y coherentes, capacidad de síntesis.

Referencias

David Litchfield (2005)

The Database Hacker's Handbook: Defending Database Servers

Kaspersky Lab (2017)

IT Security Risks Survey 2017