Laboratorio #10
Seguridad - *PAREJAS*

I. Modalidad y fecha de entrega

a) Debe ser enviado antes de la fecha límite de entrega: lunes 31 de octubre a las 05:20 PM
b) Luego de la fecha límite se restarán 10 puntos por cada hora de atraso en la entrega

II. Descripción de la actividad

El objetivo del laboratorio consiste en aplicar el principio de seguridad "menor privilegio" mediante grupos, privilegios y usuarios, para proteger la integridad de los objetos de la base de datos: la integridad en seguridad de la información se refiere a que solamente los usuarios autorizados pueden realizar cambios.

Así mismo, se busca profundizar en el concepto de amenaza y vulnerabilidad, para identificar una vulnerabilidad de un sitio Web y explotarla mediante un ataque de SQL Injection, a través de una herramienta diseñada para este propósito, a fin de comprender el ataque y aplicar las medidas necesarias en los desarrollos futuros del estudiante.

Principio del menor privilegio

El principio del menor privilegio se refiere a darle al usuario los permisos, privilegios, accesos mínimos, etc., que necesita para desarrollar sus funciones. De esta forma se previene la modificación no autorizada, por error, negligencia o a propósito.

Aunque se pueden otorgar permisos directamente a los usuarios, en una organización mediana o grande esto puede derivar rápidamente en un descontrol sobre los permisos que tienen los usuarios. Un mejor enfoque es el uso de grupos o roles en la base de datos, identificando funciones comunes entre los usuarios, de forma que los permisos mínimos se otorgan a los grupos, y luego se asigna un grupo a un usuario, asegurando el principio del menor privilegio para todos los usuarios de ese rol.

Un usuario de la base de datos no necesariamente representa un usuario operativo en una aplicación, aunque puede existir dicha correspondencia. Para otorgar y revocar privilegios se usan los comandos GRANT y REVOKE. Estos comandos pertenecen al subconjunto de DCL (Data Control Language). Investigue la sintaxis y uso de los comandos[1] para la creación de roles, asignación y des asignación de privilegios.

Actividad

Debe dejar constancia de todos los pasos realizados. Cree una base de datos que represente el sistema de asignación de cursos de la universidad. Debe contemplar las tablas de estudiantes, cursos y asignaciones. A continuación desarrolle una aplicación para el manejo de las asignaciones en Python que solicite un usuario y contraseña para conectarse a la base de datos y tenga las siguientes funcionalidades:

- Creación de estudiantes: solicita el carnet, nombres y apellidos.
- Creación del curso: solicita el código, nombre, cupo actual y cupo máximo.
- Creación de la asignación: solicita el carnet y el código del curso. Debe verificar que no se haya alcanzado el cupo máximo del curso para permitir la asignación.

Cree los siguientes grupos y al menos dos usuarios por cada grupo:

- admin_nivel1: tiene privilegios de SELECT, UPDATE e INSERT en todas las tablas creadas.
- admin_nivel2: tiene privilegios de INSERT en la tabla de estudiantes y curso
- admin_nivel3: tiene privilegios de SELECT sobre todas las tablas creadas, y INSERT sobre la tabla asignación.

[1] https://www.postgresql.org/docs/8.1/user-manag.html, https://www.postgresql.org/docs/8.0/sql-createuser.html

Realice las siguientes acciones:

- Ingrese a la aplicación con un usuario del grupo admin_nivel1, cree un estudiante, un curso, y asigne al estudiante a dicho curso
- Ingrese a la aplicación con un usuario del grupo admin_nivel2, cree cinco estudiantes y dos cursos. Intente realizar una asignación, ¿qué sucede en este caso?
- Ingrese a la aplicación con un usuario del grupo admin_nivel3, realice al menos cinco asignaciones. Intente crear un estudiante, ¿qué sucede en este caso?
- Modifique el grupo admin_nivel1, colóquele una contraseña que expire en cierto momento del día. Antes de que la contraseña expire, conéctese con un usuario de este grupo, cree un nuevo estudiante, curso, y asígnelos.
- Después que la contraseña expire, intente crear un nuevo estudiante con el mismo usuario. ¿Qué sucede en este caso?
- Modifique el grupo administradores_nivel3, u otórguele el privilegio de CREATE sobre la tabla estudiantes. Vuelva a intentar crear un estudiante, ¿qué sucede ahora?

Práctica 2. SQL Injection

Para este ejercicio utilizaremos como objeto de ataque el siguiente sitio Web: http://testphp.vulnweb.com/. Este sitio es mantenido por Acunetix, una organización que se enfoca en el desarrollo de herramientas para el escaneo de vulnerabilidades, y el propósito de este sitio es ser blanco de pruebas de seguridad. Por ética nunca debemos atacar sistemas a menos que contemos con la aprobación explícita del dueño del sistema como en este caso, para efectos de Hacking Etico.

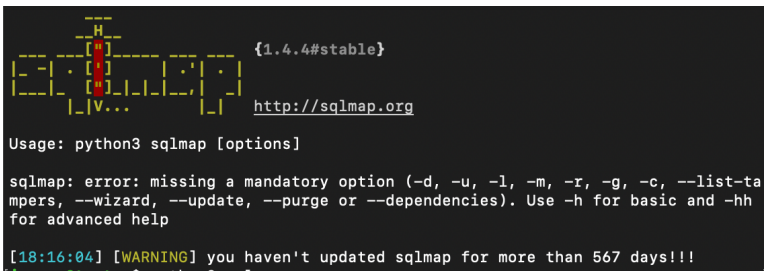
OWASP TOP 10

En el sitio del OWASP TOP 10 podemos encontrar más información sobre el ataque de inyección. Este tipo de ataques ocupó hasta el año pasado el puesto número uno, y aunque bajó al tercer lugar, sigue siendo un ataque bastante común y cuyos resultados pueden causar un gran impacto negativo en una organización.

Actividad

Comience revisando el sitio Web. Haga clic en el menú “Browse categories” y luego en la sección “Posters”. Analice la URL de la página. ¿Qué función tiene “cat=1” al final de la URL? ¿Qué sucede si cambia el valor 1 por 2? ¿Qué sucede si cambia el valor 1 por una cadena de texto? ¿Obtuvo alguna información adicional?

Descargue e instale la herramienta SQLMAP[2]. Pruebe que esté instalada correctamente ingresando el comando “sqlmap”, debe ver una pantalla similar a esta:



Escriba el comando aqlmap -h, esto mostrará la categoría “Enumeration”. Aquí se describen las opciones que puede usar para recolectar información de la base de datos.

[2] https://sqlmap.org/

```
Enumeration:
  These options can be used to enumerate the back-end database
  management system information, structure and data contained in the
  tables

  -a, --all           Retrieve everything
  -b, --banner        Retrieve DBMS banner
  --current-user      Retrieve DBMS current user
  --current-db        Retrieve DBMS current database
  --passwords         Enumerate DBMS users password hashes
  --tables            Enumerate DBMS database tables
  --columns           Enumerate DBMS database table columns
  --schema            Enumerate DBMS schema
  --dump              Dump DBMS database table entries
  --dump-all          Dump all DBMS databases tables entries
  -D DB               DBMS database to enumerate
  -T TBL              DBMS database table(s) to enumerate
  -C COL              DBMS database table column(s) to enumerate
```

A continuación, ejecute el comando sqlmap – u [URL sitio Web] (dependiendo del sistema operativo, deberá incluir la URL entre comillas). Responda las siguientes preguntas:

- ¿Qué tipo de DBMS utiliza este sitio?
- ¿Se corresponde la información obtenida con la información que obtuvo al probar ingresar una cadena de texto para el parámetro cat?
- ¿Qué versión tiene el DBMS?
- ¿A qué tipos de ataque es vulnerable el parámetro cat?

Ejecute el comando sqlmap – u [URL sitio Web] –dbs

- ¿Cuáles son los nombres de las bases de datos?

Con la información del comando -h, prepare una instrucción para obtener las tablas de la cada de una de las bases de datos identificadas. Liste las tablas. Finalmente, seleccione algunas tablas y prepare un comando para obtener más información sobre ellas, muestre los resultados obtenidos.

Temas a reforzar

- DCL
- Seguridad

III. Recursos y bibliografía

- Documentación de PostgreSQL: https://www.postgresql.org/docs/current/sql-grant.html

IV. Documentos a entregar

- Backup de la base de datos PostgreSQL con el nombre **lab10-[números de carnet].bak**
- Los programas de Python escritos para manipular la base de datos.
- Documento PDF con las capturas de pantalla de los ejercicios 1 y 2.

V. Evaluación

- Ejercicio #1: 70 puntos
- Ejercicio #2: 30 puntos
- **Total: 100 puntos**