UNIVERSIDAD TECNOLÓGICA
DE XICOTEPEC DE JUÁREZ

Plan de seguridad
“Comité de trasplantes”

NOMBRE DE EQUIPO: KICS

INTEGRANTES:

Karen Alyn Fosado Rodríguez – 210764

Carlos Martin Hernández de Jesús -210496

Sebastián Márquez García -210505

ASIGNATURA: Administración de Base De datos

DOCENTE TITULAR: Marco A. Ramírez Hernández

PROGRAMA EDUCATIVO: Ing. en Desarrollo y Gestión de Software

PERIODO: Enero – Abril 2024

Contenido

Objetivo general: 3
Objetivos Específicos: 3
Usuarios, Roles y Privilegios: 3
Usuarios: 3
Roles: 4
Privilegios: 5
Respaldo Automatizado: 6

Objetivo general:

Asegurar la integridad, confidencialidad y disponibilidad de los datos relacionados con el comité de trasplantes mediante la implementación de medidas de seguridad apropiadas en la base de datos.

Objetivos Específicos:

1. Control de Acceso:
   - Se establecerá un sistema de usuarios, roles y privilegios en la base de datos para regular el acceso según las responsabilidades del personal del comité de trasplantes.
   - Solo el personal autorizado tendrá acceso a datos sensibles relacionados con los trasplantes, como la información del paciente y los registros médicos.
2. Respaldo y Recuperación:
   - Se implementará un plan de respaldo automatizado que realice copias de seguridad de manera quincenal de la base de datos del comité de trasplantes.
   - Se garantizará la capacidad de restaurar rápidamente datos en caso de pérdida o corrupción debido a fallos del sistema o errores humanos.
3. Auditoría y Monitoreo:
   - Se utilizarán herramientas de auditoría y monitoreo para registrar y analizar la actividad de la base de datos, incluidas las consultas y modificaciones realizadas por el personal autorizado.
   - Se establecerán alertas para detectar y responder a actividades sospechosas que puedan indicar intentos de acceso no autorizado o violaciones de seguridad.
4. Cumplimiento Legal y Regulatorio:
   - Se garantizará el cumplimiento de la protección de datos de salud y la privacidad del paciente.
   - Se establecerán lineamientos y procedimientos para manejar incidentes de seguridad y violaciones de datos de acuerdo con las leyes y regulaciones pertinentes.

Usuarios, Roles y Privilegios:

Usuarios:

- Se crearán usuarios específicos para diferentes roles dentro del proyecto, como médicos y pacientes del comité de trasplantes.
- Cada usuario tendrá credenciales de acceso únicas para acceder a la base de datos.

- SQL

```
/*Crear usuarios prueba  */
CREATE USER 'medico'@'localhost' IDENTIFIED BY 'contraseña';
CREATE USER 'paciente'@'localhost' IDENTIFIED BY 'contraseña';
```

- NoSQL

```
test> use bd_hospital_kics

      db.createUser({
          user: "medico",
          pwd: "contraseña_medico",
          roles: [{ role: "readWrite", db: "bd_hospital_kics" }]
      })

      db.createUser({
          user: "paciente_donador",
          pwd: "contraseña_donador",
          roles: [{ role: "readWrite", db: "bd_hospital_kics" }]
      })

      db.createUser({
          user: "paciente_donatario",
          pwd: "contraseña_donatario",
          roles: [{ role: "readWrite", db: "bd_hospital_kics" }]
      })
```

Roles:

- Se crearán roles específicos, como "Médico" y "Paciente" para definir las responsabilidades y privilegios de cada usuario.
- Los roles se asignarán según las funciones y responsabilidades del personal del proyecto.

- SQL

```
/* Crear roles */
CREATE ROLE 'medico_role';
CREATE ROLE 'paciente_role';
```

- NoSQL

```
bd_hospital_kics> db.createRole({
                      role: "medico_role",
                      privileges: [
                          { resource: { db: "bd_hospital_kics", collection: "" }, actions: ["find", "insert", "update", "remove"] }
                      ],
                      roles: []
                  })

                  db.createRole({
                      role: "paciente_donador_role",
                      privileges: [
                          { resource: { db: "bd_hospital_kics", collection: "" }, actions: ["find", "insert", "update", "remove"] }
                      ],
                      roles: []
                  })

                  db.createRole({
                      role: "paciente_donatario_role",
                      privileges: [
                          { resource: { db: "bd_hospital_kics", collection: "" }, actions: ["find", "insert", "update", "remove"] }
                      ],
                      roles: []
                  })
```

```
bd_hospital_kics> db.grantRolesToUser("medico", [{ role: "medico_role", db: "bd_hospital_kics" }])
                  db.grantRolesToUser("paciente_donador", [{ role: "paciente_donador_role", db: "bd_hospital_kics" }])
                  db.grantRolesToUser("paciente_donatario", [{ role: "paciente_donatario_role", db: "bd_hospital_kics" }])
```

Privilegios:

- Se otorgarán privilegios específicos a cada rol para acceder y manipular los datos de la base de datos según sea necesario.
- Los privilegios se asignarán de manera que el personal tenga acceso solo a los datos relevantes para sus funciones.

- SQL

```
/* Asignar privilegios a los roles */
GRANT SELECT, INSERT, UPDATE, DELETE ON bd_hospital_210764.* TO 'medico_role';
GRANT SELECT ON bd_hospital_210764.* TO 'paciente_role';


/* *Asignar roles a los usuarios:*/
GRANT medico_role TO 'medico'@'localhost';
GRANT paciente_role TO 'paciente'@'localhost';
```

- NoSQL

```
bd_hospital_kics> db.createRole({
                      role: "medico_role",
                      privileges: [
                          { resource: { db: "bd_hospital_kics", collection: "" }, actions: ["find", "insert", "update", "remove"] }
                      ],
                      roles: []
                  })

                  db.createRole({
                      role: "paciente_donador_role",
                      privileges: [
                          { resource: { db: "bd_hospital_kics", collection: "" }, actions: ["find", "insert", "update", "remove"] }
                      ],
                      roles: []
                  })

                  db.createRole({
                      role: "paciente_donatario_role",
                      privileges: [
                          { resource: { db: "bd_hospital_kics", collection: "" }, actions: ["find", "insert", "update", "remove"] }
                      ],
                      roles: []
                  })
```

Respaldo Automatizado:

- Se implementará un plan de respaldo automatizado que incluirá:
    - Frecuencia: Se realizarán copias de seguridad quincenales de la base de datos del comité de trasplantes.
    - Métodos: Se utilizarán herramientas y scripts automatizados para realizar y programar los respaldos de manera regular.
    - Almacenamiento: Las copias de seguridad se almacenarán en ubicaciones seguras fuera del servidor de la base de datos para garantizar la recuperación en caso de fallo del sistema.

El plan de seguridad consiste en proporcionar una estructura clara para implementar y mantener medidas de seguridad efectivas en la base de datos del proyecto de comité de trasplantes, garantizando así la protección adecuada de la información confidencial y sensible.

0 0 */15 * * mysqldump -u [usuario] -p[contraseña] [nombre_basedatos] > /ruta/para/guardar/respaldo.sql

En este ejemplo:

- 0 0 */15 * * especifica que la tarea se ejecute a la medianoche (00:00) de cada 15 días (el día exacto se determina automáticamente).
- mysqldump es el comando para realizar el respaldo.
- -u [usuario] especifica el nombre de usuario de la base de datos.
- -p[contraseña] especifica la contraseña del usuario (sin espacio entre -p y la contraseña).
- [nombre_basedatos] es el nombre de la base de datos que deseas respaldar

- > /ruta/para/guardar/respaldo.sql redirige la salida del comando mysqldump a un archivo llamado respaldo.sql, ubicado en la ruta especificada.

La razón principal para programar respaldos automáticos cada 15 días en lugar de con mayor frecuencia es encontrar un equilibrio entre la frecuencia de los respaldos y el impacto en el rendimiento del sistema y el almacenamiento.