Seguridad informática

Universidad Galileo de Guatemala
Técnico en Desarrollo de Software
Seguridad Informática
Ingeniero Randy Fernando Juárez Najarro

ACTIVIDAD 8

ALMACENAMIENTO Y AUTENTICACIÓN DE CONTRASEÑAS

Brenda Samara Escobar Avila

Carnet: 23005735

Guatemala, lunes 09 de diciembre de 2024

INTRODUCCIÓN

El almacenamiento y la autenticación de contraseñas son elementos críticos en la seguridad de sistemas informáticos y plataformas digitales. En un mundo donde la cantidad de datos personales y financieros que se manejan en línea crece exponencialmente, la protección de las credenciales de usuario es esencial para evitar accesos no autorizados y potenciales ataques cibernéticos. Las contraseñas, aunque siguen siendo uno de los métodos más comunes de autenticación, son inherentemente vulnerables si no se manejan de forma adecuada. Por ello, tanto el almacenamiento seguro como la implementación de métodos eficaces de autenticación son aspectos que no pueden subestimarse.

La autenticación, por otro lado, es el proceso mediante el cual se verifica que un usuario es quien dice ser. El método más común es el uso de contraseñas, pero también existen métodos adicionales como la autenticación de dos factores (2FA) y la autenticación biométrica. El 2FA, por ejemplo, añade una capa extra de seguridad al requerir una segunda forma de verificación, como un código enviado a un dispositivo móvil o un token generado por una aplicación. Esta medida hace que, incluso si una contraseña es comprometida, el acceso al sistema aún esté protegido por una segunda barrera.

El almacenamiento y la autenticación de contraseñas son, sin duda, pilares en la seguridad de la información y la protección de la privacidad digital. La implementación de prácticas seguras y el uso de tecnologías adecuadas son imprescindibles para mitigar los riesgos y garantizar la integridad de los sistemas y la confianza de los usuarios.

ALMACENAMIENTO Y AUTENTICACIÓN DE CONTRASEÑAS

Instrucciones:

Aplicar la arquitectura de **almacenamiento y autenticación** de contraseñas vistas en la unidad 6.

1. Puede utilizar cualquier lenguaje de programación y cualquier librería adicional.
2. Coloque las pantallas mostrando el proceso y su funcionamiento. Describa qué parte del proceso ocurre en cada imagen, ya sea del código o del funcionamiento.

- **Justificar por qué esta arquitectura es segura, que factores posee que la acreditan para ser considerada una práctica segura.**
- **Responder a la pregunta y justificar ¿Aplicar esta arquitectura sustituye el utilizar doble factor de autenticación (2FA)?**

REPOSITORIO EN GIT

https://github.com/SamyER33/23005735_A8_Si.git

Sistema de Registro y Autenticación de Usuarios con MongoDB y Python

Este proyecto es un sistema básico de registro y autenticación de usuarios, desarrollado en Python. Utiliza **MongoDB Atlas** como base de datos para almacenar de manera segura los datos de los usuarios, incluyendo las contraseñas encriptadas mediante `bcrypt`.

Características

- **Registro de usuarios**: Permite registrar nuevos usuarios almacenando sus credenciales de manera segura.
- **Autenticación de usuarios**: Verifica las credenciales ingresadas contra la base de datos.
- **Cifrado de contraseñas**: Usa `bcrypt` para garantizar la seguridad de las contraseñas.
- **Conexión con MongoDB Atlas**: Se conecta a una base de datos MongoDB en la nube para almacenar los datos.

Requisitos Previos

Antes de ejecutar el código, asegúrate de tener lo siguiente:

1. **Python 3.7 o superior** instalado.

2. Las siguientes bibliotecas instaladas:

- `pymongo`: Para interactuar con MongoDB.
- `bcrypt`: Para el hash y verificación de contraseñas.
- Puedes instalarlas ejecutando:

```
bash
pip install pymongo bcrypt
```

3. Una cuenta en [MongoDB Atlas] (https://www.mongodb.com/cloud/atlas) con un clúster configurado y accesible desde tu dirección IP.

Configuración

1. Actualizar la conexión de MongoDB:

```Python
client =
MongoClient("mongodb+srv://<usuario>:<contraseña>@<cluster>.mongodb.net/?retryWrites=true&w=majority&appName=<nombre-app>")
```

- Usa tus credenciales y detalles del clúster.

Uso del Sistema

Opciones en el Menú

1. Registrar usuario:

- Ingresa un nombre de usuario único.
- Ingresa una contraseña que será cifrada antes de almacenarse.

2. Iniciar sesión:

- Ingresa tu nombre de usuario y contraseña.\
- El sistema verificará las credenciales y permitirá el acceso si son correctas.

3. Salir:

- Cierra el programa.

Estructura del Código

1. Conexión a MongoDB:

- Se conecta a MongoDB Atlas usando la biblioteca `pymongo`.

2. Funciones principales:

- `register_user(username, password)`: Crea un nuevo usuario y almacena su contraseña en formato hash.
- `authenticate_user(username, password)`: Verifica las credenciales del usuario ingresado.

3. Menú interactivo:

- Proporciona opciones para registrar usuarios, iniciar sesión o salir.

Seguridad

- **Contraseñas cifradas**: `bcrypt` se usa para generar hashes únicos por contraseña.
- **Verificación segura**: Los hashes se verifican sin exponer las contraseñas originales.
- **Base de datos remota**: Los datos se almacenan en MongoDB Atlas, que ofrece cifrado en tránsito y en reposo.

Notas Importantes

- Asegúrate de agregar tu dirección IP en la configuración de Network Access de MongoDB Atlas para permitir el acceso.
- Considera usar variables de entorno para almacenar credenciales de conexión, en lugar de incluirlas directamente en el código.

Seguridad informática

ile Edit Selection View Go Run Terminal Help Actividad8

flaskTarea.py

flaskTarea.py > main

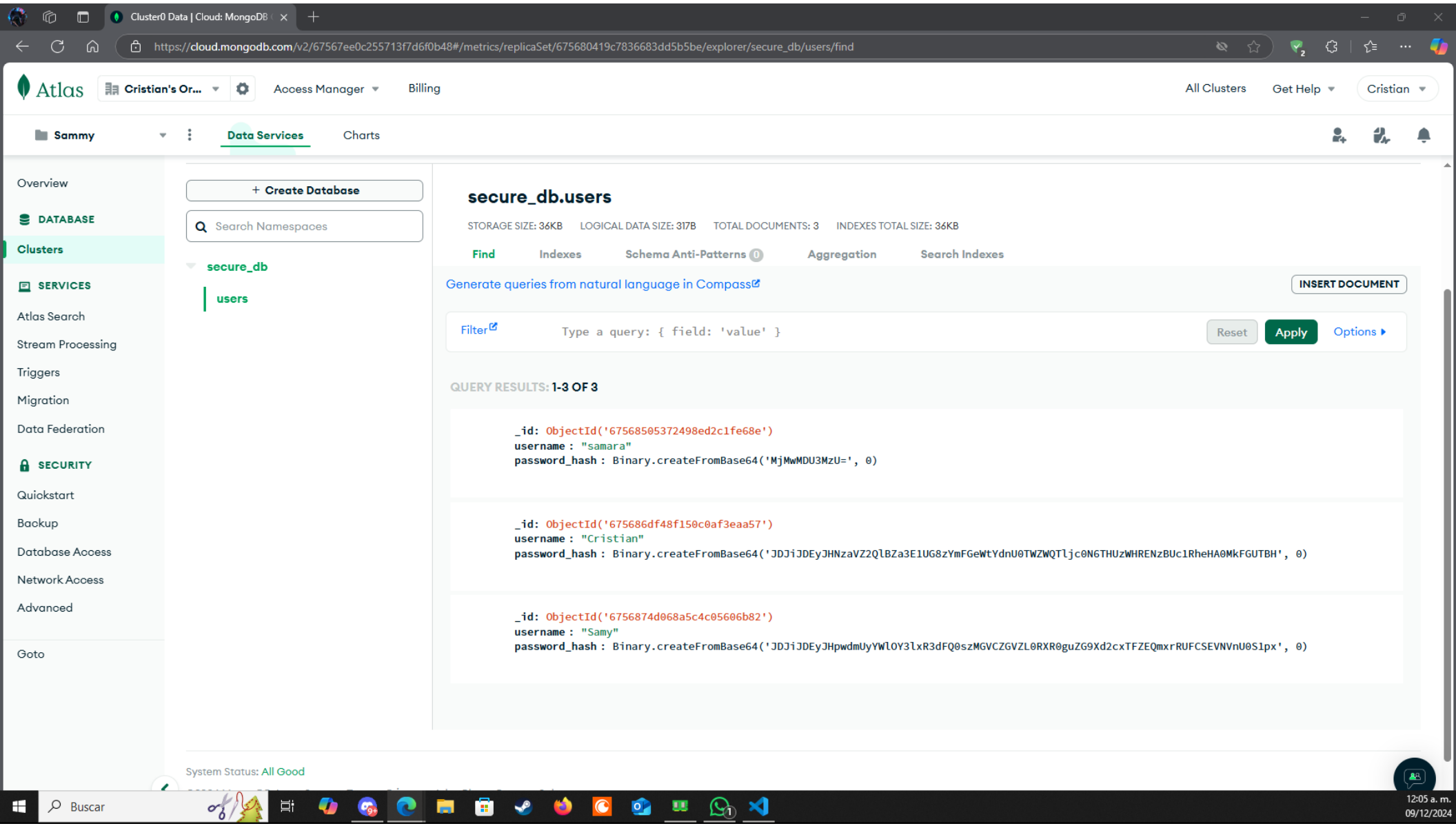
```python
import bcrypt
from pymongo import MongoClient

# Conexión a MongoDB Atlas
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE Python

```
e object have been deprecated. Please switch to next_update_utc.
  if value.this_update <= _datetime.utcnow() < value.next_update:
1
Ingrese un nombre de usuario: Samy
Ingrese una contraseña: 5735
Usuario 'Samy' registrado con éxito.

=== Sistema de Registro y Login ===
1. Registrar usuario
2. Iniciar sesión
3. Salir
Seleccione una opción: 2
Ingrese su nombre de usuario: Samy
Ingrese su contraseña: 5735
Inicio de sesión exitoso para el usuario 'Samy'.

=== Sistema de Registro y Login ===
1. Registrar usuario
2. Iniciar sesión
3. Salir
Seleccione una opción: 2
Ingrese su nombre de usuario: Samy
Ingrese su contraseña: 5736
Error: Credenciales inválidas para el usuario 'Samy'.

=== Sistema de Registro y Login ===
1. Registrar usuario
2. Iniciar sesión
3. Salir
Seleccione una opción:
```

Buscar

12:03 a. m.
09/12/2024


Cluster0 Data | Cloud: MongoDB
https://cloud.mongodb.com/v2/67567ee0c255713f7d6f0b48#/metrics/replicaSet/675680419c7836683dd5b5be/explorer/secure_db/users/find
Atlas
Cristian's Or...
Access Manager
Billing
All Clusters
Get Help
Cristian
Sammy
Data Services
Charts
Overview
DATABASE
Clusters
SERVICES
Atlas Search
Stream Processing
Triggers
Migration
Data Federation
SECURITY
Quickstart
Backup
Database Access
Network Access
Advanced
Goto
+ Create Database
Search Namespaces
secure_db
users
secure_db.users
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 317B
TOTAL DOCUMENTS: 3
INDEXES TOTAL SIZE: 36KB
Find
Indexes
Schema Anti-Patterns 0
Aggregation
Search Indexes
Generate queries from natural language in Compass
INSERT DOCUMENT
Filter
Type a query: { field: 'value' }
Reset
Apply
Options
QUERY RESULTS: 1-3 OF 3
_id: ObjectId('67568505372498ed2c1fe68e')
username : "samara"
password_hash : Binary.createFromBase64('MjMwMDU3MzU=', 0)
_id: ObjectId('675686df48f150c0af3eaa57')
username : "Cristian"
password_hash : Binary.createFromBase64('JDJiJDEyJHNzaVZ2QlBZa3E1UG8zYmFGeWtYdnU0TWZWQTljc0N6THUzWHRENzBUc1RheHA0MkFGUTBH', 0)
_id: ObjectId('6756874d068a5c4c05606b82')
username : "Samy"
password_hash : Binary.createFromBase64('JDJiJDEyJHpwdmUyYWlOY3lxR3dFQ0szMGVCZGVZL0RXR0guZG9Xd2cxTFZEQmxrRUFCSEVNVnU0S1px', 0)
System Status: All Good
Buscar
12:05 a. m.
09/12/2024

JUSTIFICACIÓN DE SEGURIDAD

1. **Por qué esta arquitectura es segura:**

   - **Uso de bcrypt:** Proporciona hashes de contraseñas robustas, resistentes a ataques de fuerza bruta debiro al “salting” y su diseño computacionalmente costoso.

   - **HTTPS (TLS/SSL):** Garantiza que los datos transmitidos estén cifrados, protegiendo las credenciales en tránsito.

   - **TDE (Transparent Data Encryption):** Protege la base de datos de acceso no autorizado si el disco o la base de datos son comprometidos.

2. **¿Reemplaza al 2FA?**

   No, esta arquitectura no sustituye al 2FA. Mientras que asegura el almacenamiento y transmisión de contraseñas, el 2FA agrega una capa adicional de seguridad al requerir un factor externo al usuario. Es altamente recomendable usar ambas estrategias.