گزارش فنی: سامانه API RESTful امن با ذخیره‌سازی رمزنگاری‌شده اطلاعات و مدیریت کلید

۱ شرح معماری کلی سامانه

سامانه مورد نظر یک API RESTful امن است که با هدف ارائه خدمات احراز هویت، مدیریت امن داده‌های حساس و مدیریت کلیدهای رمزنگاری طراحی و پیاده‌سازی شده است. این سامانه با استفاده از چارچوب FastAPI توسعه یافته که به دلیل کارایی بالا، پشتیبانی از برنامه‌نویسی ناهمزمان و تولید مستندات خودکار ،OpenAPI به عنوان ابزار اصلی انتخاب شده است. برای ذخیره‌سازی داده‌ها، از پایگاه داده MongoDB استفاده شده که یک پایگاه داده NoSQL انعطاف‌پذیر است و امکان مقیاس‌پذیری افقی و مدیریت داده‌های ساختاریافته و نیمه‌ساختاریافته را فراهم می‌کند. همچنین، مدیریت کلیدهای رمزنگاری به صورت ایزوله و امن با بهره‌گیری از Vault HashiCorp انجام می‌شود که یک ابزار استاندارد صنعتی برای مدیریت اسرار و کلیدها به شمار می‌رود.

سامانه از طریق Compose Docker اجرا می‌شود که استقرار و مدیریت سرویس‌های مختلف را در کانتینرهای جداگانه تسهیل می‌کند و سازگاری بین محیط‌های توسعه و تولید را تضمین می‌کند. معماری سامانه شامل سه بخش اصلی است:

- **برنامه FastAPI**: این بخش مسئول دریافت و پردازش درخواست‌های ،HTTP احراز هویت کاربران با استفاده از توکن‌های ،JWT رمزنگاری و رمزگشایی داده‌های حساس با کلیدهای اختصاصی هر کاربر و تعامل با پایگاه داده MongoDB و Vault HashiCorp است.
- **پایگاه داده MongoDB**: این پایگاه داده برای ذخیره‌سازی اطلاعات کاربران (مانند نام کاربری، رمز عبور هش‌شده، salt و اطلاعات احراز هویت دو مرحله‌ای) و داده‌های حساس رمزنگاری‌شده (شامل نوع داده و مقدار رمزنگاری‌شده) استفاده می‌شود.
- **Vault HashiCorp**: این ابزار مدیریت امن کلیدهای رمزنگاری را بر عهده دارد و شامل کلید اصلی (Master Key) و کلیدهای منحصر به فرد هر کاربر است که با کلید اصلی رمزنگاری شده‌اند.

این معماری تضمین می‌کند که داده‌های حساس در تمامی مراحل انتقال و ذخیره‌سازی رمزنگاری شده باقی بمانند و کلیدهای رمزنگاری به صورت جداگانه و با امنیت بالا مدیریت شوند.

۲ الگوریتم‌های رمزنگاری و هش استفاده شده

برای تأمین امنیت داده‌ها و احراز هویت کاربران، از الگوریتم‌های رمزنگاری و هش استاندارد استفاده شده است:

- **هش رمز عبور**: الگوریتم bcrypt با ضریب کار ۱۲ برای هش کردن رمزهای عبور به کار گرفته شده است که مقاومت بالایی در برابر حملات brute-force ارائه می‌دهد. برای هر کاربر، یک salt منحصر به فرد ۱۶ بایتی با استفاده از تابع `secrets.token_hex(16)` تولید می‌شود تا از حملات table rainbow جلوگیری شود. علاوه بر این، یک pepper سراسری که در فایل `env.` ذخیره شده، به رمز عبور افزوده می‌شود تا لایه امنیتی بیشتری ایجاد کند. خروجی نهایی در فیلد `hashed_password` ذخیره می‌شود.
- **رمزنگاری داده‌های حساس**: از الگوریتم AES-۲۵۶-GCM استفاده شده که یک الگوریتم رمزنگاری متقارن با قابلیت احراز اصالت است. هر کاربر دارای یک کلید ۲۵۶ بیتی منحصر به فرد است که با کلید

اصلی رمزنگاری شده و در Vault ذخیره می‌شود. برای هر عملیات رمزنگاری، یک nonce منحصر به فرد ۱۲ بایتی با `secrets.token_bytes(12)` تولید می‌شود تا از تکرار ciphertext جلوگیری شود.

- **امضای توکن JWT**: توکن‌های JWT با الگوریتم HS۲۵۶ (HMAC-SHA۲۵۶) امضا می‌شوند و زمان انقضای آن‌ها ۱ ساعت تعیین شده است. کلید امضا (`JWT_SECRET`) به صورت امن در فایل `.env` نگهداری می‌شود.

این الگوریتم‌ها به دلیل امنیت بالا، استاندارد بودن و پشتیبانی گسترده در کتابخانه‌های رمزنگاری انتخاب شده‌اند و تضمین می‌کنند که رمزهای عبور غیرقابل بازیابی، داده‌های حساس به صورت امن رمزنگاری‌شده و توکن‌های احراز هویت در برابر دستکاری ایمن باشند.

۳ ساختار پایگاه داده

پایگاه داده MongoDB با نام `secure_api` شامل دو مجموعه اصلی است که به شرح زیر طراحی شده‌اند:

- **مجموعه `users`**: این مجموعه شامل اطلاعات کاربران است و فیلدهای زیر را در بر می‌گیرد:
  - `_id`: شناسه یکتا (ObjectId).
  - `username`: نام کاربری منحصر به فرد.
  - `email`: آدرس ایمیل برای احراز هویت دو مرحله‌ای.
  - `hashed_password`: هش bcrypt رمز عبور همراه با salt و pepper.
  - `salt`: salt منحصر به فرد ۱۶ بایتی.
  - `failed_attempts`: تعداد تلاش‌های ناموفق ورود (حداکثر ۵ تلاش).
  - `last_failed_attempt_time`: زمان آخرین تلاش ناموفق برای اعمال قفل ۱۵ دقیقه‌ای.
  - `totp_secret`: راز base۳۲ برای TOTP در احراز هویت دو مرحله‌ای.
  - `two_factor_enabled`: وضعیت فعال بودن احراز هویت دو مرحله‌ای.
- **مجموعه `sensitive_data`**: این مجموعه برای ذخیره‌سازی داده‌های حساس رمزنگاری‌شده طراحی شده و شامل فیلدهای زیر است:
  - `_id`: شناسه یکتا (ObjectId).
  - `user_id`: شناسه کاربر که به `_id` در مجموعه `users` ارجاع می‌دهد.
  - `data_type`: نوع داده حساس (مانند "card_number").
  - `encrypted_value`: داده رمزنگاری‌شده به صورت hex-encoded (شامل nonce و cipher-text).

این ساختار، جداسازی داده‌های احراز هویت و داده‌های حساس را تضمین می‌کند و از ذخیره‌سازی امن و رمزنگاری‌شده داده‌ها اطمینان می‌دهد.

۴ روش مدیریت کلید

مدیریت کلیدها با استفاده از Vault HashiCorp به صورت زیر انجام می‌شود:

- **کلید اصلی (Master Key)**: یک کلید AES-۲۵۶ است که در مسیر `kv/master_key` در Vault ذخیره می‌شود. این کلید برای رمزنگاری و رمزگشایی کلیدهای کاربران استفاده می‌شود و در صورت عدم وجود، به صورت خودکار تولید و ذخیره می‌گردد.
- **کلیدهای کاربران**: هر کاربر یک کلید AES-۲۵۶ منحصر به فرد دارد که در زمان ثبت‌نام تولید شده، با کلید اصلی رمزنگاری می‌شود و همراه با nonce در مسیر `kv/user_keys/<username>` در Vault ذخیره می‌گردد.
- **چرخش کلید**: از طریق endpoint `/rotate-master-key`، فرآیند چرخش کلید اجرا می‌شود که شامل مراحل زیر است:
  ۱. بازیابی کلید اصلی قدیمی.
  ۲. تولید کلید اصلی جدید.
  ۳. رمزگشایی کلید هر کاربر با کلید قدیمی و رمزنگاری مجدد آن با کلید جدید.
  ۴. به‌روزرسانی کلید اصلی در Vault.

این روش، امنیت کلیدها را تضمین کرده و امکان چرخش کلیدها را بدون نیاز به تغییر داده‌های حساس فراهم می‌کند.

۵ دلایل انتخاب ابزارها و روش‌ها

ابزارها و روش‌های مورد استفاده بر اساس معیارهای زیر انتخاب شده‌اند:

- **FastAPI**: به دلیل کارایی بالا، پشتیبانی از برنامه‌نویسی ناهمزمان، مستندات خودکار OpenAPI و اعتبارسنجی داده‌ها با Pydantic، برای توسعه API امن و مقیاس‌پذیر انتخاب شد.
- **MongoDB**: به عنوان یک پایگاه داده NoSQL، انعطاف‌پذیری در مدیریت داده‌های ساختاریافته و نیمه‌ساختاریافته را فراهم کرده و با کتابخانه `motor` به صورت ناهمزمان با FastAPI یکپارچه می‌شود.
- **Vault HashiCorp**: به دلیل ارائه امنیت بالا، قابلیت مدیریت دسترسی و چرخش کلید به عنوان یک ابزار استاندارد صنعتی انتخاب شد.
- **Compose Docker**: برای استقرار آسان و مدیریت سرویس‌های چند کانتینری و تضمین سازگاری محیط‌های توسعه و تولید استفاده می‌شود.
- **کتابخانه Cryptography**: به دلیل پیاده‌سازی امن و آزمایش‌شده الگوریتم‌های رمزنگاری مانند AES-۲۵۶-GCM، برای تضمین امنیت داده‌ها به کار گرفته شد.

این ترکیب از ابزارها و روش‌ها، سیستمی امن، مقیاس‌پذیر و قابل نگهداری را فراهم می‌کند که تمامی نیازهای پروژه را برآورده می‌سازد.