INFINITYPALS PayPal combo list checker

1. Project Objective

Develop a Golang-based PayPal combo checker that include building and API for automated account verification, ensuring it bypasses security measures like captchas and rate limits. The API must be optimized for high-speed processing and seamless integration with an admin panel for user management, licensing, and security control.

2. Reverse Engineering & Security Bypass

- Intercept and analyze authentication requests using Burp Suite or Fiddler.

- Identify required headers, tokens, session cookies, and response structures.

- Extract and manage CSRF tokens and session handling mechanisms.

- Implement NextCaptcha and CAK (Captcha AI Kit) for captcha solving.

- Rotate residential proxies to avoid detection and IP bans.

3. API Development in Golang

- Use Gin or Fiber for fast request handling.

- Implement multi-threading with goroutines to handle large-scale requests.

- Integrate retry mechanisms for failed requests due to captchas or rate limits.

- Randomize headers, user-agents, and session data to mimic real user behavior.

API Endpoints:

- POST /check → Accepts credentials, processes authentication, and returns captured details.

- GET /status → Checks API health and request limits.

Output Data (JSON Format):

- Full Name

- Country

- Linked Cards (Brand, Last Digits)

- Linked Banks (Issuer, Type, Last Digits)

4. Admin Panel Features

- Subscription & License Key Management: Generate, track, and revoke license keys.

- User Management: View user activity, ban or unban users, and monitor logs.

- Security Features: Implement IP whitelisting or blacklisting and brute-force protection.

- Software Authentication & Protection:

- Require valid license keys for access.

- Bind keys to specific devices (hardware locking).

- Prevent key sharing with device registration.

- Block access after multiple failed authentication attempts.

- Admin Settings & Logs: Backup and restore functionality, error logging, and debugging tools.

5. Development & Testing Steps

- Finalize API structure and authentication handling.

- Implement and test the API in Golang with security bypass techniques.

- Integrate the API with the admin panel.

- Optimize for performance, security, and scalability.

Also build a good looking front end for the project (check picture)

Expl of the project :

Combo: mail:pass

luis.svalle@yahoo.com:Guason12.

Checker check the validity of the account if he is valid he capture the data needed final result should be like that :

luis.svalle@yahoo.com:Guason12. | Full Name: [aurora rafael cirilo] | Country: [Mexico] | Card Issuers: [MASTER\_CARD - Brand: (Mastercard)] | Card Last Digits: [••07] | Bank Issuers: [Null] | Bank Last Digits: [Null]

Bank and card = Null if no cc or bank account is linked

Hits must be saved on a ‘hits’ folder and classified by countries