Go API Gateway Security Packages Reference

Authentication & Authorization

github.com/golang-jwt/jwt/v5

Creates, signs, and validates JSON Web Tokens (JWT) for stateless authentication. Handles token parsing, claims validation, and supports multiple signing algorithms (HMAC, RSA, ECDSA). Essential for securing API endpoints without maintaining server-side session state.

golang.org/x/crypto/bcrypt

Securely hashes passwords using the bcrypt algorithm with automatic salt generation. Computationally expensive by design to prevent brute-force attacks. Used for storing user credentials safely in databases.

github.com/casbin/casbin/v2

Policy-based access control engine that supports multiple authorization models (ACL, RBAC, ABAC). Separates authorization logic from business code. Allows complex permission rules like "user X can perform action Y on resource Z under condition W."

Rate Limiting

golang.org/x/time/rate

Token bucket rate limiter from Go's extended library. Allows controlling request rates with burst capacity. Simple and efficient for basic rate limiting per IP or API key to prevent abuse and DDoS attacks.

github.com/ulule/limiter/v3

Feature-rich rate limiting with multiple storage backends (memory, Redis, etc.). Supports per-route limits, distributed rate limiting across multiple gateway instances, and custom key extraction. Better for production environments with multiple servers.

github.com/throttled/throttled/v2

HTTP rate limiter with support for quotas and different rate limiting strategies. Provides middleware for easy integration. Good for implementing tiered API plans (free tier: 100 req/hour, paid tier: 10000 req/hour).

HTTP Framework & Routing

github.com/gin-gonic/gin

High-performance HTTP web framework with middleware support, JSON validation, and routing. Provides convenient error handling and context management. Popular choice for building REST APIs with minimal boilerplate.

github.com/gorilla/mux

Powerful URL router and dispatcher. More lightweight than Gin. Excellent for complex routing patterns, path variables, and matching based on HTTP methods, headers, or query parameters.

github.com/rs/cors

Handles Cross-Origin Resource Sharing (CORS) configuration. Controls which domains can access your API from browsers. Prevents unauthorized cross-domain requests while allowing legitimate frontend applications to communicate with your gateway.

Security Middleware

github.com/unrolled/secure

Implements security best practices through middleware. Automatically adds security headers (HSTS, CSP, X-Frame-Options), enforces HTTPS, prevents clickjacking, and protects against common web vulnerabilities. One-stop solution for HTTP security headers.

Input Validation

github.com/go-playground/validator/v10

Struct and field validation based on tags. Validates request payloads against rules (required fields, email format, length constraints, numeric ranges). Prevents malformed data from reaching your business logic and protects against injection attacks.

Logging & Tracing

github.com/google/uuid

Generates universally unique identifiers for request tracking. Assign each incoming request a unique ID to trace it through distributed systems, microservices, and logs. Essential for debugging and monitoring in production.

go.uber.org/zap

High-performance structured logging library. Creates machine-readable JSON logs with fields (request_id, user_id, status_code). Much faster than standard library logging and better for production monitoring and alerting systems.

github.com/rs/zerolog

Zero-allocation JSON logger focused on performance. Similar to Zap but with a different API. Provides structured logging without impacting request latency. Choose between Zap or Zerolog based on preference.

Configuration Management

github.com/joho/godotenv

Loads environment variables from .env files. Keeps sensitive configuration (API keys, database credentials, JWT secrets) out of source code. Essential for secure deployment practices and managing different environments (dev, staging, prod).

Reverse Proxy & Load Balancing

net/http/httputil (Standard Library)

Built-in reverse proxy functionality. Forwards requests from gateway to backend services, handles response streaming, and modifies requests/responses. Core component for implementing gateway routing to microservices.

Reliability Patterns

github.com/sony/gobreaker

Circuit breaker pattern implementation. Prevents cascading failures by stopping requests to failing services temporarily. When a backend service is down, the circuit breaker "opens" and returns errors immediately instead of waiting for timeouts, protecting your gateway from overload.

Monitoring & Metrics

github.com/prometheus/client_golang

Exposes application metrics in Prometheus format. Tracks request counts, latencies, error rates, active connections. Essential for monitoring gateway health, setting up alerts, and creating dashboards with Grafana.

TLS/Security (Standard Library)

crypto/tls (Standard Library)

Configures TLS settings for HTTPS connections. Set minimum TLS versions, cipher suites, and certificate handling. Critical for encrypting traffic between clients and gateway, and between gateway and backend services.

Context Management (Standard Library)

context (Standard Library)

Manages request cancellation, timeouts, and deadline propagation across API boundaries. When a client disconnects, context cancellation stops downstream processing, saving resources. Essential for preventing resource leaks and controlling long-running operations.

Quick Decision Guide

Minimal Gateway: Gorilla Mux + JWT + golang.org/x/time/rate + Standard library TLS **Production Gateway**: Gin + JWT + Ulule Limiter + Zap + Secure + Casbin + Circuit Breaker + Prometheus **High Security**: All of the above + Validator + CORS + proper Context timeout handling