Política de Seguridad de la Plataforma de Microservicios

1. Propósito

Definir los lineamientos y controles de seguridad para el diseño, desarrollo, operación y mantenimiento de los microservicios que conforman la solución, de manera que se minimicen los riesgos asociados a amenazas internas y externas. Se busca proteger los datos de los usuarios, prevenir accesos no autorizados y asegurar la continuidad operativa.

2. Alcance

Esta política aplica a todos los componentes, microservicios, versiones, entornos (desarrollo, pruebas, producción) y personal involucrado en el ciclo de vida de las siguientes aplicaciones:

- Servicio Cuentas (/cuentas)

- Servicio Reportes (/cuentas/reportes)

- Servicio KYC (/kyc)

- Servicio Apuestas (/apuestas)

- Servicio Usuarios (/users)

- Servicio Pagos (/pay)

- Servicio Juegos (/juegos)

- Servicio Sanciones (/sanciones)

Incluye tanto las API RESTful desarrolladas en Django (Python) como cualquier otro componente (bases de datos MySQL, front-ends, colas de mensajes, etc.).

3. Definiciones

- Activo de Información: Cualquier dato, base de datos, código fuente o infraestructura que maneja o procesa información sensible.

- Microservicio: Componente autónomo que expone operaciones específicas a través de endpoints.

- Usuario: Persona o sistema autorizado a consumir las APIs.

- Administrador: Personal técnico que gestiona despliegues, configuración y monitoreo de la plataforma.

- Cliente: Consumidor de los servicios que realiza operaciones (crear usuario, realizar apuesta, recargar cuenta).

4. Roles y Responsabilidades

1. Responsable de Seguridad (CISO o equivalente)

- Definir y revisar periódicamente la política de seguridad.

- Coordinar auditorías internas y externas.

- Validar cumplimiento de estándares (ISO 27001, OWASP, GDPR si aplica).

2. Equipo de Desarrollo

- Implementar prácticas de desarrollo seguro.

- Asegurar que el código cumpla con validaciones de entrada, saneamiento y manejo seguro de credenciales.

- Reportar vulnerabilidades detectadas durante el ciclo de vida del desarrollo.

3. Equipo de Operaciones / DevOps

- Configurar correctamente los servidores y contenedores.

- Gestionar actualizaciones de dependencias, parches de seguridad y backups regulares.

- Monitorear logs y alertar sobre actividades sospechosas.

4. Administrador de Base de Datos

- Asegurar la configuración de la base de datos MySQL con cuentas de acceso restringido, cifrado en reposo y en tránsito.

- Realizar respaldos periódicos y pruebas de restauración.

5. Equipo de QA / Pruebas

- Ejecutar pruebas de penetración (pentests) sobre cada endpoint.

- Validar que los controles de acceso y autenticación funcionen conforme a lo especificado.

5. Clasificación de Activos

Activo | Tipo | Nivel de Sensibilidad | Protección Aplicable

Base de datos de Cuentas | Datos Personales | Alta | Cifrado en reposo (AES-256), control de acceso basado en roles (RBAC), auditoría de accesos

Base de datos de Apuestas | Datos Transaccionales | Alta | Cifrado en tránsito (TLS 1.2+), registro de transacciones, separación de entornos dev/prod

Endpoint /kyc (Datos de identificación) | Datos Personales | Muy Alta | Autenticación fuerte (OAuth2/JWT), validación estricta de documentos, almacenamiento mínimo, eliminación de datos luego del proceso

Código fuente de cada microservicio | Propiedad Intelectual | Media | Repositorio privado, firma de commits, revisión de código, escaneo SAST

Configuraciones en settings.py (Secrets) | Credenciales | Muy Alta | Manejo a través de variables de entorno / vault, rotación periódica, acceso controlado

Infraestructura en la nube (servidores) | Infraestructura | Media-Alta | Seguridad en red (firewalls, VPNs), segmentación de VPC, monitoreo de puertos

6. Control de Acceso y Gestión de Identidades

- Principio de Mínimo Privilegio: permisos estrictamente necesarios para cada microservicio y usuario.

- OAuth2 con JWT: el microservicio de Usuarios (/users) emite tokens; roles Cliente, Staff, Administrador.

- Gestión de Contraseñas: bcrypt, políticas de longitud y complejidad, expiración cada 180 días.

- Manejo de Sesiones y Tokens: JWT de 15 minutos, refresh tokens de 7 días, bloqueo de sesiones sospechosas.

7. Desarrollo Seguro (Secure SDLC)

- Revisión de Código y Pull Requests: análisis estático (Bandit, SonarQube), rechazo de PRs con vulnerabilidades.

- Validación de Entradas: usar formularios de Django o serializers de DRF, protección contra XSS y CSRF.

- Gestión de Dependencias: mantener requirements.txt o Pipfile.lock, escaneo de vulnerabilidades trimestral.

- Pruebas de Penetración: pruebas semestrales contra OWASP Top 10.

8. Seguridad en las Comunicaciones y Red

- Cifrado en Tránsito: HTTPS (TLS 1.2+), renovación automática de certificados, deshabilitar protocolos inseguros.

- Segmentación de Red y Firewalls: subredes privadas, API Gateway, rules de Security Groups, IDS/IPS.

- Protección DDoS y Rate Limiting: limitación de tasa, WAF, planes de mitigación DDoS.

- Seguridad de Infraestructura: imágenes Docker actualizadas, parches OS semanales, acceso SSH con llaves y 2FA.

9. Gestión de Vulnerabilidades y Parches

- Escaneo Continuo: herramientas para contenedores (trivy, Dockle), reportes mensuales.

- Plan de Parches: clasificar por severidad (Críticas ≤ 48h, Altas ≤ 7 días, Medias/Bajas mensual), documentar en change log.

10. Monitoreo y Registro (Logging)

- Registro de Eventos: logs estructurados (JSON) con timestamp UTC, IP de origen, endpoint, usuario, trace ID.

- Almacenamiento Seguro de Logs: centralizar en ELK, retener 90 días, registros inmutables.

- Monitoreo y Alertas: alertas por intentos fallidos de login, patrones anómalos, uso elevado de recursos, revisión diaria de operaciones.

11. Gestión de Incidentes de Seguridad

- Detección y Clasificación: describir incidente, hora, sistemas afectados, vector de ataque.

- Respuesta Inmediata: equipo CSIRT (líder, analista forense, ingeniero de seguridad), aislar componentes, plan de recuperación.

- Comunicación y Notificación: notificar a gerencia TI y partes afectadas, autoridades si hay datos personales expuestos.

- Análisis Post-Incidencia y Mejora Continua: informe post-mortem, actualizar política y controles.

12. Continuidad de Negocio y Recuperación ante Desastres

- Backup y Recuperación: respaldos diarios, pruebas de restauración trimestrales.

- Alta Disponibilidad: despliegue multi-AZ, balanceador de carga con health checks.

- Plan de Recuperación (DRP): documentar escenarios críticos, roles, tiempos RPO/RTO.

13. Cumplimiento y Auditoría

- Normativas y Estándares: ISO/IEC 27001, OWASP Top 10, LFPDPPP.

- Auditorías Internas y Externas: internas anuales, externas cada dos años, revisión de política anual.

- Reporte de Cumplimiento: repositorio de evidencias, reuniones trimestrales con gerencia.

14. Revisión y Actualización de la Política

- Revisar cada 12 meses o tras un incidente significativo.

- Responsable de seguridad convocará equipos para validar cambios y asegurar cumplimiento.