Informe OWASP

Análisis de Amenazas

Índice

Índice....2
INFORME INICIAL....3
1. Identificación de agentes maliciosos y vectores de ataque....3
1.1 Actores maliciosos identificados....3
1.2 Vectores de ataque principales....4
2. Aplicación del OWASP Top 10 (2024)....6
A01:2024 – Broken Access Control....6
A02:2024 – Cryptographic Failures....7
A03:2024 – Injection (incluye SQLi y Prompt Injection)....7
A04:2024 – Insecure Design....7
A05:2024 – Security Misconfiguration....8
A06:2024 – Vulnerable and Outdated Components....8
A07:2024 – Identification and Authentication Failures....8
A08:2024 – Software and Data Integrity Failures....9
A09:2024 – Security Logging and Monitoring Failures....9
A10:2024 – Server-Side Request Forgery (SSRF) y API Abuse....9
3. Evaluación del impacto técnico y de negocio....10
3.1 Impacto Técnico....10
3.2 Impacto de Negocio....11
4. Conclusión....12

Informe inicial

Threat Model — Plataforma de comercio online con Chatbot IA
Aplicación del OWASP Top 10 (2024) y Análisis de Amenazas
Autor: Equipo de Ciberseguridad

1. Identificación de agentes maliciosos y vectores de ataque

1.1 Actores maliciosos identificados

Actor	Descripción	Motivación	Nivel Riesgo
Atacante externo	Usuario anónimo intentando explotar vulnerabilidades del frontend, backend o chatbot	Robo de datos, impacto reputacional, sabotaje	Alto
Usuario legítimo malintencionado	Alguien con cuenta válida que intenta escalar privilegios o acceder a datos de RRHH o Marketing	Curiosidad, beneficio personal, filtración	Alto
Bots automatizados	Scripts o herramientas de reconocimiento y explotación	Recolección masiva de datos, explotación	Medio
Desarrollador con errores involuntarios	Generación de fallos de seguridad debido a configuraciones inseguras o malas prácticas	Vulnerabilidades inadvertidas	Medio
Amenaza interna	Acceso no autorizado a la base de datos, backups o infraestructura	Exfiltración o manipulación de datos	Alto

1.2 Vectores de ataque principales

Frontend (S3/CloudFront):

- Riesgos: inyección de scripts (XSS), archivos JS manipulados, exposición de endpoints públicos.
- Ejemplos: scripts maliciosos en la UI, tokens almacenados en cliente, CORS mal configurados.
- Mitigaciones inmediatas: forzar HTTPS/TLS + HSTS; aplicar Content Security Policy (CSP) y Sub Resource Integrity (SRI); revisar y restringir CORS; proteger buckets S3 (no públicos).
- Prioridad: Alta

Manipulación de tráfico / comunicaciones

- Riesgos: interceptación o modificación de tráfico (MITM) si no hay TLS o certificados débiles.
- Ejemplos: sniffing en redes públicas, downgrade de TLS.
- Mitigaciones inmediatas: TLS 1.2+/1.3 obligatorio, rotación y gestión automática de certificados, aplicar pinning en clientes críticos.
- Prioridad: Alta

Backends (EC2 o Lambda):

- Riesgos: SQL Injection, endpoints internos expuestos, autenticación insuficiente, gestión incorrecta de sesiones/tokens.
- Ejemplos: queries concatenadas, endpoints sin auth, tokens sin expiración o revocación.
- Mitigaciones inmediatas: consultas parametrizadas/ORM, políticas de autenticación robustas (JWT con expiración y revocación), WAF y rate-limiting, revisión de endpoints expuestos.
- Prioridad: Alta

Chatbot:

- Riesgos: prompt injection, acceso no autorizado a datos internos por mala configuración, divulgación involuntaria de información sensible.

- Ejemplos: usuario que induce al modelo a revelar datos, mediador que ejecuta consultas a la BD sin validación.
- Mitigaciones inmediatas: introducir un mediador que limite y sanitice el contexto enviado al modelo; no enviar campos sensibles salvo autorización explícita; filtros de salida (detección de PII); auditoría de llamadas al modelo y a la base de datos.
- Prioridad: Alta

Base de datos (RDS PostgreSQL):

- Riesgos: acceso no autorizado, escalado de privilegios, fuga de datos sensibles por falta de cifrado o validaciones insuficientes.
- Ejemplos: RDS expuesto a Internet, usuario de aplicación con permisos excesivos, snapshots accesibles.
- Mitigaciones inmediatas: no exponer RDS públicamente; cifrado en reposo via KMS y en tránsito; usuarios DB con permisos mínimos; prepared statements; controlar acceso a snapshots/backups.
- Prioridad: Alta

AWS (Infraestructura):

- Riesgos: IAM mal configurado (privilegios excesivos), buckets S3 públicos, exposición de puertos/servicios sin protección.
- Ejemplos: políticas con wildcard (*), keys estáticas en repos, Security Groups muy permisivos.
- Mitigaciones inmediatas: revisar y aplicar principio de mínimo privilegio; usar roles temporales (STS); desactivar buckets públicos; securizar Security Groups y usar subnets privadas / bastion; habilitar CloudTrail y alertas para cambios IAM.
- Prioridad: Alta

SSRF / API Abuse:

- Riesgos: APIs que permiten peticiones a recursos internos (metadata, servicios internos) o abuso de cuota.
- Ejemplos: endpoints que aceptan URLs y hacen requests sin validación; tokens extraídos vía metadata service.
- Mitigaciones inmediatas: validar y normalizar URLs; deny list de metadata/internal IPs; limitar egress; aplicar cuotas y rate-limits; usar proxies controlados.
- Prioridad: Alta

2. Aplicación del OWASP Top 10 (2024)

El OWASP Top 10 de 2024 incluye categorías orientadas a entornos de aplicaciones modernas, APIs y modelos de IA. La siguiente revisión adapta estas categorías al proyecto del chatbot y la infraestructura en AWS.

A01:2024 – Broken Access Control

- Añadir: Owner (Backend/DB), Prioridad: Alta.
- Implementación: RBAC a nivel API y DB; políticas IAM con least privilege; checks server-side en cada endpoint (no confiar en cliente).
- Pruebas/aceptación: pruebas de IDOR/authorization fuzzing, SAST rules para verificar checks de autorización.
- Detección: alertas por accesos a tablas sensibles fuera de horario o por cuentas no autorizadas.
- Ejemplo: no dar al servicio chatbot permisos SELECT sobre tablas RRHH; usar vistas controladas con columnas mínimas.

A02:2024 – Cryptographic Failures

- Añadir: Owner (Infra/DevOps), Prioridad: Inmediata.

- Implementación: TLS 1.2+ (preferible 1.3), HSTS, certificados automatizados (ACME), RDS + S3 SSE-KMS, KMS key rotation policy.
- Contraseñas: bcrypt/argon2 con salt; nunca almacenar secretos en repos.
- Pruebas/aceptación: escaneo TLS (SSL Labs), verificación de cifrado en reposo en snapshots y backups.
- Métrica: % conexiones TLS, % volúmenes cifrados.

A03:2024 – Injection (incluye SQLi y Prompt Injection)

- Añadir: Owner (Backend + ML Ops), Prioridad: Alta.
- SQL: usar prepared statements/ORM, parámetros, whitelist input validation, least-privilege DB user.
- Prompt injection: mediador entre UI y modelo, strip/escape de instrucciones meta, no incluir datos sensibles en contexto por defecto.
- Pruebas: DAST/DAST personalizado para prompt injection (fuzzing de prompts), pentest con casos de prompt injection.
- Detección: patrones en respuestas que contienen PII o comandos inesperados.

A04:2024 – Insecure Design

- Añadir: Owner (Arquitectura), Prioridad: Alta.
- Implementación: threat modeling por feature, separación de datos sensibles, diseño por capas y principios de Zero Trust.
- Artefacto: diagrama de datos y flujos (qué puede ver el chatbot, qué no).
- Validación: revisión de diseño en PRs y gating de seguridad.

A05:2024 – Security Misconfiguration

- Añadir: Owner (Infra), Prioridad: Inmediata.
- Implementación: IaC scanning (tfsec/checkov), políticas que bloqueen S3 público, Security Groups restrictivos, no exponer RDS.
- Pruebas: escaneo de configuraciones periódicas, guardrails con AWS Config / SCP en Org.
- Detección: alertas de Security Hub/GuardDuty por recursos públicos.

A06:2024 – Vulnerable and Outdated Components

- Añadir: Owner (DevSecOps), Prioridad: Media–Alta.
- Implementación: SCA (Dependabot/Snyk), bloqueo de merge si CVE críticos, imágenes de contenedor escaneadas y firmadas.
- Pruebas: builds que fallen por CVEs críticos; calendar de actualizaciones.
- Métrica: tiempo promedio de parcheo (days to patch).

A07:2024 – Identification and Authentication Failures

- Añadir: Owner (Auth), Prioridad: Inmediata.
- Implementación: JWT con algoritmo fuerte (RS256), expiraciones cortas, refresh tokens con revocación, MFA para admins, session management (revocación y logout global).
- Pruebas: atacar flujos de autenticación, brute-force detection, pruebas de token replay.
- Detección: múltiples intentos fallidos, tokens usados desde múltiples IPs.

A08:2024 – Software and Data Integrity Failures

- Añadir: Owner (DevOps/ML Ops), Prioridad: Media.
- Implementación: firmar artefactos y modelos (Sigstore/cosign), reproducible builds, checksums en despliegues, control de acceso a pipelines.
- Para ML: control de versiones del dataset y del modelo, registro de origen de datos, validación de integridad antes de uso.
- Pruebas: verificación de firmas antes de deploy; pruebas de integridad de modelos.

A09:2024 – Security Logging and Monitoring Failures

- Añadir: Owner (SecOps), Prioridad: Inmediata.
- Implementación: centralizar logs (CloudWatch → SIEM), habilitar CloudTrail-data events para S3/RDS, audit logs de DB, retention definida.
- Detección: playbooks, alertas para accesos a tablas sensibles, detección de anomalías (baselining).
- Métrica: MTTD / MTTR objetivos, % servicios con logging habilitado.

A10:2024 – Server-Side Request Forgery (SSRF) y API Abuse

- Añadir: Owner (Backend/Infra), Prioridad: Alta.
- Implementación: validar URLs, denylist de metadata/internal IPs (169.254.169.254), limitar egress, usar proxy de salida, quotas y rate limiting por API/key.
- Pruebas: DAST que incluya SSRF, pentesting de endpoints que aceptan URLs.

- Detección: requests salientes a 169.254.169.254 o a range internas, egress inusual.

3. Evaluación del impacto técnico y de negocio

3.1 Impacto Técnico

Ataque	Consecuencia Técnica	Severidad	Mitigación inmediata
Fuga de datos sensibles de RRHH o usuarios	Exposición completa o parcial de información personal (PII) y posibles filtraciones públicas	Alta	Revocar credenciales comprometidas, auditar accesos, aplicar cifrado en reposo (KMS) y en tránsito; notificar al DPO si procede.
SQL Injection	Modificación, borrado o exfiltración de datos críticos; posible escalado a control total de BD	Alta	Revisar y parametrizar queries (prepared statements/ORM), aplicar WAF, ejecutar pentest/DAST sobre endpoints críticos.
Exposición de endpoints	Acceso no autorizado a APIs o servicios internos; pivoting interno	Alta	Revisar exposición de endpoints, aplicar autorización en cada endpoint, usar API Gateway/WAF, poner endpoints internos en subnets privadas.
Vulneración del chatbot	Filtración de información sensible o ejecución de peticiones indebidas (prompt injection / exfiltración)	Alta	Introducir mediador que límite contexto, sanitice prompts y audite consultas; filtros de salida para PII.

Configuraciones AWS inseguras	Compromiso de infraestructura (buckets públicos, IAM excesivo, SG permisivos)	Alta	Reforzar IAM (least privilege), bloquear S3 público, revisar Security Groups, habilitar CloudTrail/GuardDuty y remediar hallazgos críticos.

3.2 Impacto de Negocio

Riesgo	Impacto	Probabilidad	Mitigación / nota de negocio
Filtración de datos	Impacto reputacional, pérdida de confianza de clientes, sanciones legales (GDPR, etc.)	Alta	Preparar plan de comunicación, notificación a autoridades (si aplica), revisar DLP y gobernanza de datos.
Caída del sistema	Interrupción del servicio del chatbot y la plataforma → pérdida de ventas y productividad	Media–Alta	SLA/DR plan, alta disponibilidad (multi-AZ), pruebas regulares de restore y runbooks de recuperación.
Manipulación de datos	Decisiones erróneas en RRHH o Marketing, daño operativo	Media	Controles de integridad, backups verificados, logging/auditoría y separación de entornos.
Abuso de APIs	Costes inesperados, uso fraudulento de recursos, impacto operativo	Media	Implementar cuotas, rate-limiting, facturación por clave, uso anómalo

Configuraciones incorrectas en AWS	Pérdida de control del sistema, acceso no autorizado a recursos sensibles	Alta	Revisión periódica de IaC, políticas de guardrails (AWS Config/SCP), formación a equipos y auditorías regulares.

4. Conclusión

El análisis basado en OWASP Top 10 (2024) identifica como riesgos principales para la plataforma: acceso indebido a datos sensibles por fallos de control de acceso; inyecciones (especialmente SQL Injection y prompt injection hacia el chatbot); configuraciones inseguras en la infraestructura AWS (IAM, S3, RDS, Security Groups); fallos de autenticación y gestión de tokens en las APIs; y exposición de endpoints internos que podrían ser manipulados por el chatbot o atacantes.

Para mitigar estos riesgos es imprescindible aplicar seguridad desde el diseño: implementación estricta de roles y least-privilege, cifrado en tránsito y en reposo, validación y sanitización rigurosa de peticiones (incluyendo prompts hacia el modelo), mediador para el chatbot que limite contexto y audite consultas, y revisiones continuas de la configuración AWS. Además, debe establecerse un programa operativo que incluya pruebas (SAST/DAST/pentest con pruebas específicas de IA), monitorización y alertas centralizadas, runbooks de respuesta a incidentes y métricas (MTTD/MTTR, % de endpoints protegidos, % de secretos rotados).

Los mayores riesgos son control de acceso deficiente, inyecciones (SQL y prompt), y configuraciones AWS inseguras; priorizar controles de acceso, mediación del chatbot, cifrado y monitoreo permitirá reducir significativamente la exposición y el impacto operativo/legislativo.