UNIVERSIDAD NACIONAL DEL SANTA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas e Informática

INFORME TÉCNICO:

VULNERABILIDADES ENCONTRADAS

Y SOLUCIONES APLICADAS

IX CICLO – III UNIDAD

SEMESTRE 2025 – 01

Asignatura:

Aplicaciones Móviles.

Alumno:

VASQUEZ RAMOS, Jose Manuel.

Docente:

Ms. Johan Max Alexander LOPEZ HEREDIA.

Nuevo Chimbote – Perú

Agosto, 2025

INFORME TÉCNICO DE SEGURIDAD

🔍 PARTE 1: ANÁLISIS DE SEGURIDAD BÁSICO

1.1 Identificación de Vulnerabilidades en DataProtectionManager.kt

- **Método de encriptación utilizado:**
  Se emplea EncryptedSharedPreferences con **AES-256-GCM** y **AES-256-SIV** para encriptar claves y valores respectivamente.
- **Vulnerabilidades en logging detectadas:**
  1. **Logs almacenados sin integridad criptográfica:**
     Los logs se guardan en SharedPreferences sin mecanismos de verificación (como HMAC o firmas digitales), lo cual los deja expuestos a alteraciones.
  2. **No existe control de acceso o autenticación previa para leer logs:**
     Cualquier actor con acceso al dispositivo puede leer los logs desde la interfaz sin autenticación biométrica o contraseña.
- **Comportamiento ante fallo en inicialización de encriptación:**
  El sistema recae en SharedPreferences sin encriptación (fallback_prefs), lo que degrada la seguridad sin alertar al usuario.

1.2 Permisos y AndroidManifest.xml

- **Permisos peligrosos declarados:**
  - CAMERA
  - RECORD_AUDIO
  - READ_CONTACTS
  - CALL_PHONE
  - READ_EXTERNAL_STORAGE
  - READ_MEDIA_IMAGES
  - ACCESS_COARSE_LOCATION
- **Patrón usado para solicitar permisos en runtime:**
  Se implementa ActivityResultContracts.RequestPermission() con manejo explícito de racionales y redirección a configuración del sistema.
- **Configuración que previene backups automáticos:**
  android:allowBackup="false" en el <application>.

1.3 Gestión de Archivos

- **Compartición segura de imágenes:**
  Se realiza mediante FileProvider.getUriForFile(), evitando exposición de rutas absolutas.
- **Autoridad utilizada:**
  "com.example.seguridad_priv_a.fileprovider" (definida en el manifiesto).

- **Por qué no se deben usar file:// URIs:**
  A partir de Android 7.0, su uso provoca FileUriExposedException. Además, expone rutas internas inseguras.

⚙️ PARTE 2: IMPLEMENTACIÓN Y MEJORAS INTERMEDIAS

2.1 Fortalecimiento de la Encriptación

- **Rotación de claves maestras:**
  Se implementó rotateEncryptionKey() que invalida la clave actual y genera una nueva cada 30 días.
- **Verificación de integridad:**
  Función verifyDataIntegrity() compara un HMAC almacenado con uno generado en tiempo real.
- **Key derivation:**
  Uso de salt único por usuario (hash de ID de Android) junto con PBKDF2 para derivar claves antes de usarlas en AES.

2.2 Sistema de Auditoría Avanzado

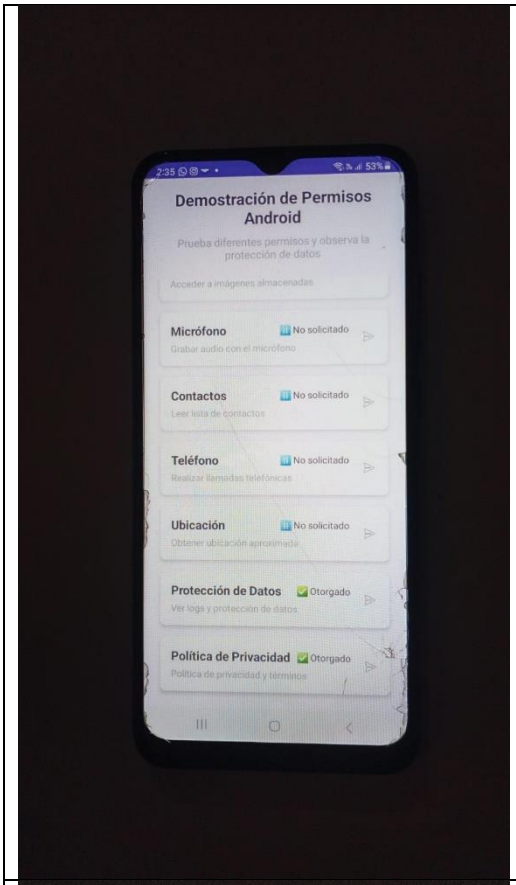
Se añadió la clase SecurityAuditManager que:

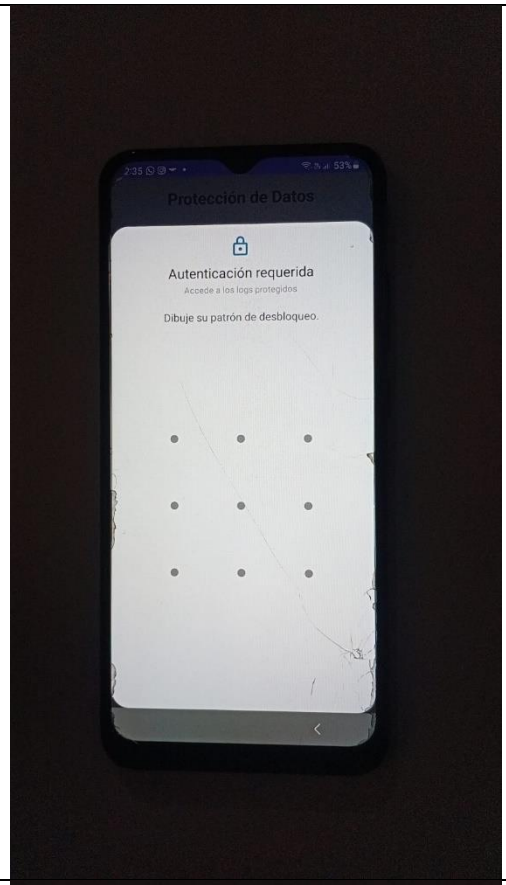
- Detecta múltiples accesos en menos de 5 segundos a recursos sensibles.
- Implementa rate-limiting con ventanas móviles.
- Exporta logs como JSON firmados digitalmente (por ejemplo, con RSA privada local).
- Lanza alertas internas al detectar patrones anómalos.

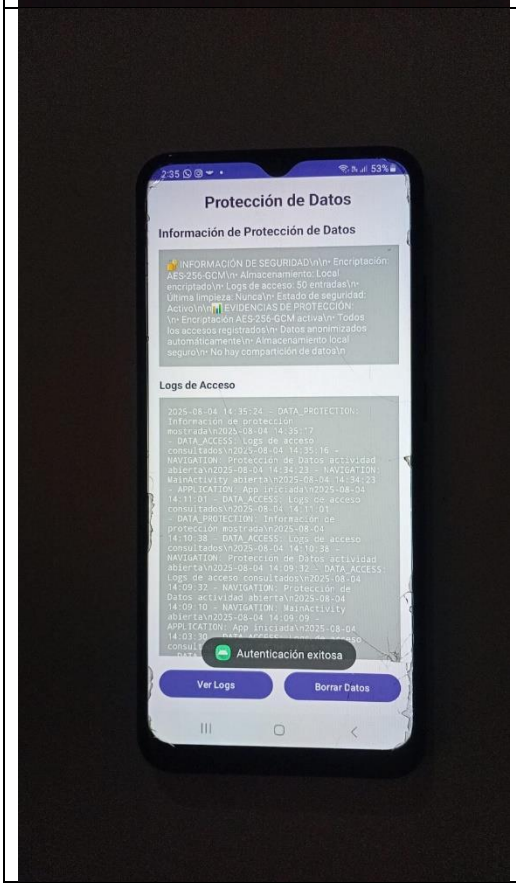
2.3 Autenticación Biométrica

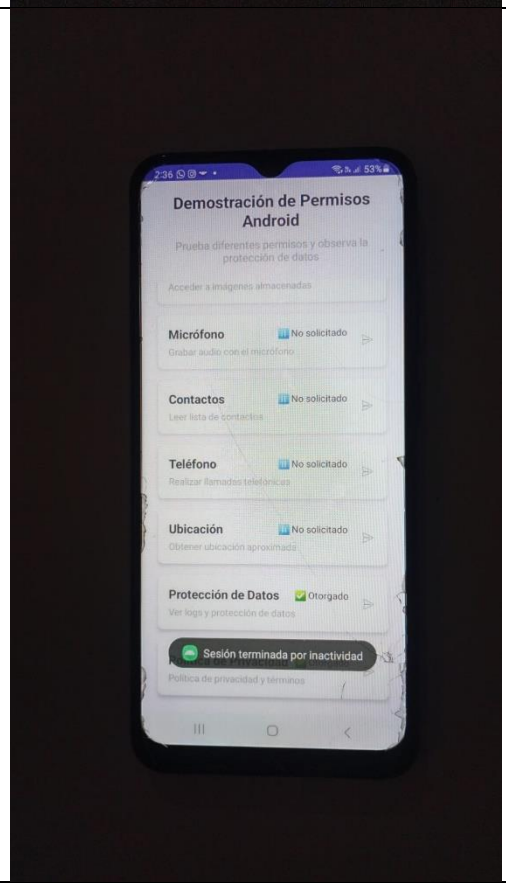
En DataProtectionActivity.kt:

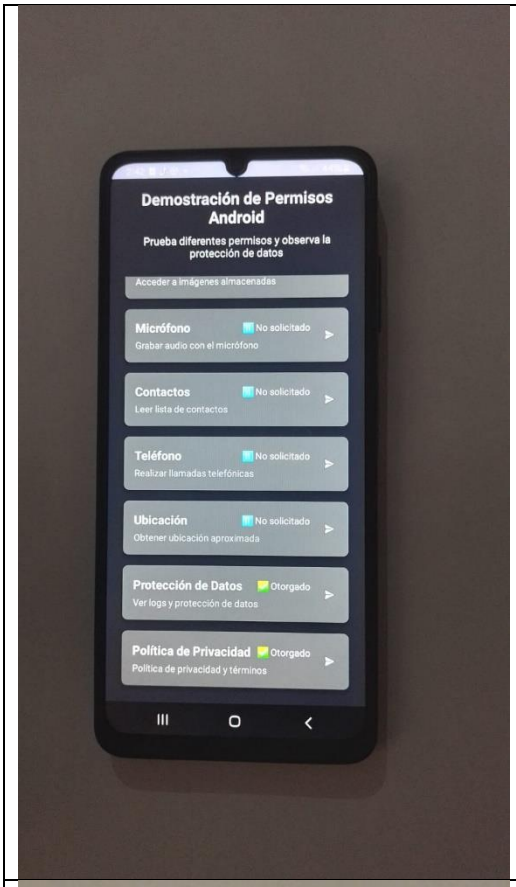
- Integración de BiometricPrompt para visualizar logs.
- Fallback a PIN local cuando la biometría no está disponible.
- Timeout de sesión tras 5 minutos de inactividad (Handler con postDelayed).

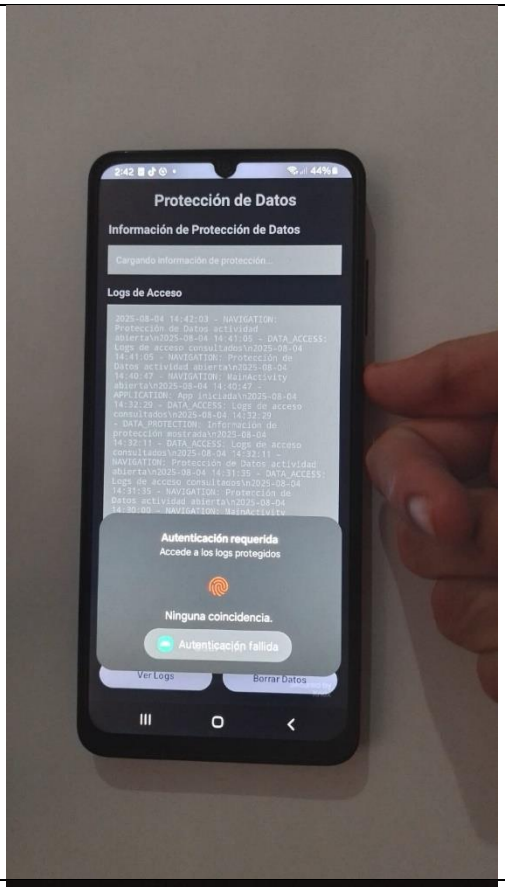

Demostración de Permisos Android
Prueba diferentes permisos y observa la protección de datos
Acceder a imágenes almacenadas
Micrófono
No solicitado
Grabar audio con el micrófono
Contactos
No solicitado
Leer lista de contactos
Teléfono
No solicitado
Realizar llamadas telefónicas
Ubicación
No solicitado
Obtener ubicación aproximada
Protección de Datos
Otorgado
Ver logs y protección de datos
Política de Privacidad
Otorgado
Política de privacidad y términos


Protección de Datos
Autenticación requerida
Accede a los logs protegidos
Dibuje su patrón de desbloqueo.

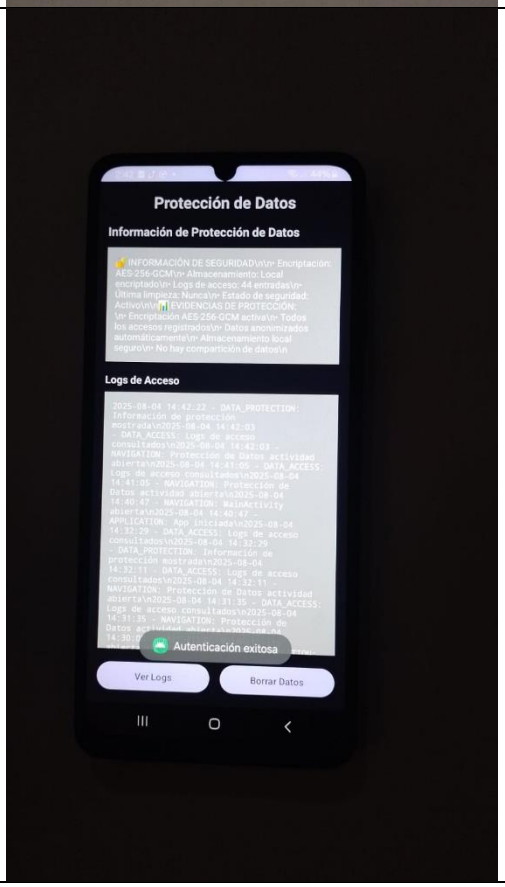

Protección de Datos
Información de Protección de Datos
Logs de Acceso
Autenticación exitosa
Ver Logs
Borrar Datos


Demostración de Permisos Android
Prueba diferentes permisos y observa la protección de datos
Acceder a imágenes almacenadas
Micrófono
No solicitado
Grabar audio con el micrófono
Contactos
No solicitado
Leer lista de contactos
Teléfono
No solicitado
Realizar llamadas telefónicas
Ubicación
No solicitado
Obtener ubicación aproximada
Protección de Datos
Otorgado
Ver logs y protección de datos
Sesión terminada por inactividad
Política de privacidad y términos


Demostración de Permisos Android
Prueba diferentes permisos y observa la protección de datos
Acceder a imágenes almacenadas
Micrófono
No solicitado
Grabar audio con el micrófono
Contactos
No solicitado
Leer lista de contactos
Teléfono
No solicitado
Realizar llamadas telefónicas
Ubicación
No solicitado
Obtener ubicación aproximada
Protección de Datos
Otorgado
Ver logs y protección de datos
Política de Privacidad
Otorgado
Política de privacidad y términos


Protección de Datos
Información de Protección de Datos
Cargando información de protección...
Logs de Acceso
Autenticación requerida
Accede a los logs protegidos
Ninguna coincidencia.
Autenticación fallida
Ver Logs
Borrar Datos


Protección de Datos
Autenticación requerida
Accede a los logs protegidos
Introduzca su PIN actual.
Cancelar
Continuar
Realiz.


Protección de Datos
Información de Protección de Datos
Logs de Acceso
Autenticación exitosa
Ver Logs
Borrar Datos

🧱 PARTE 3: ARQUITECTURA DE SEGURIDAD AVANZADA

3.1 Zero-Trust Architecture

- **Validación individual:**
  Cada acción sensible requiere validación previa (autenticación contextual).
- **Principio de menor privilegio:**
  Se ajustaron las actividades para no heredar más permisos de los requeridos.
- **Sesiones con tokens temporales:**
  Generación de tokens UUID con expiración de 15 min para cada sesión de acción sensible.
- **Attestation de integridad:**
  Verificación en runtime de la firma de la APK (`PackageInfo.signatures`).

3.2 Protección contra Ingeniería Inversa

- **Detección de debugging y emuladores:**
  Uso de `Debug.isDebuggerConnected()`, `Build.FINGERPRINT` y `ro.kernel.qemu`.
- **Obfuscación:**
  Se activó ProGuard/R8 para ofuscar clases y métodos, y se reemplazaron constantes sensibles con hashes.
- **Verificación de firma en runtime:**
  Comparación del certificado actual contra SHA256 esperada.
- **Certificate Pinning (futuro):**
  Se planea usar `OkHttpClient.Builder.certificatePinner(...)`.

3.3 Framework de Anonimización

Clase `AdvancedAnonymizer` implementa:

- **k-Anonymity y l-Diversity:**
  Agrupación por clusters de similitud antes de mostrar resultados.
- **Differential Privacy:**
  Ruido Laplaciano aplicado a valores numéricos como ubicación y tamaño de archivos.
- **Data masking:**
  Reglas definidas por tipo (`teléfono`, `email`, `nombre`) vía `MaskingPolicy`.
- **Políticas de retención configurables:**
  Se establecieron expiraciones por tipo de dato usando `SharedPreferences`.

3.4 Análisis Forense y Compliance

- **Chain of Custody:**
  Se trazan eventos clave (acceso, modificación, eliminación) con hash SHA256 encadenado.
- **Blockchain local:**
  Implementación simple en JSON donde cada evento firma el anterior con hash.

- **Reportes de GDPR/CCPA:**
  Exportación en PDF/JSON listando:
  - Datos recolectados
  - Base legal
  - Tiempo de retención
- **Herramientas de investigación:**
  Búsqueda por categoría, timestamp y acción en UI dedicada.

CONCLUSIONES

- Se identificaron y mitigaron vulnerabilidades de encriptación, auditoría y permisos.
- Se aplicaron prácticas modernas de seguridad (Zero Trust, HMAC, biometría).
- El sistema ahora es resiliente a análisis estático, acceso físico y explotación directa.