Guía de Rotación de Credenciales Comprometidas

⚠️ Urgencia: ALTA PRIORIDAD

Si has identificado que las credenciales de tu aplicación pueden haber sido expuestas en el repositorio de Git, es CRÍTICO que sigas estos pasos inmediatamente.

🔒 Credenciales que Deben Rotarse

1. Stripe

¿Por qué es crítico?
Stripe maneja pagos y datos financieros sensibles. Una clave comprometida permite:
- Acceso a información de clientes
- Creación de cargos fraudulentos
- Reembolsos no autorizados

Pasos de Rotación:

1. **Acceder al Dashboard de Stripe**
   - Ir a: https://dashboard.stripe.com/apikeys
   - Iniciar sesión con tus credenciales

2. **Generar Nuevas Claves**
   - Hacer clic en “Create secret key”
   - Copiar la nueva clave secreta (solo se muestra una vez)
   - Hacer clic en “Create publishable key” si es necesario
   - Copiar la nueva clave pública

3. **Actualizar Claves en la Aplicación**
   ```
   bash
   # En tu archivo .env
   STRIPE_SECRET_KEY=sk_live_NUEVA_CLAVE_AQUI
   NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=pk_live_NUEVA_CLAVE_AQUI
   ```

4. **Revocar Claves Antiguas**
   - En el dashboard de Stripe, hacer clic en “… ” junto a la clave antigua
   - Seleccionar “Roll key ” o “Delete”
   - Confirmar la revocación

5. **Verificar que Todo Funciona**
   - Probar un pago de prueba
   - Verificar que no hay errores en los logs

---

2. Redsys

¿Por qué es crítico?
Redsys es un método de pago español ampliamente usado. Una clave comprometida permite:
- Procesamiento de pagos fraudulentos
- Acceso a transacciones de clientes

Pasos de Rotación:

1. **Contactar a tu Banco/Redsys**
   - Redsys no tiene un panel de autoservicio para rotar claves
   - Debes contactar a tu banco o al soporte de Redsys
   - Teléfono de soporte: +34 91 270 81 00
   - Email: soporte@redsys.es

2. **Solicitar Nuevas Credenciales**
   - Merchant Code (FUC)
   - Terminal Number
   - Secret Key (Clave Secreta)

3. **Actualizar Credenciales en la Aplicación**
   ```bash
   # En tu archivo .env
   REDSYS_MERCHANT_CODE=NUEVO_CODIGO
   REDSYS_TERMINAL=NUEVO_TERMINAL
   REDSYS_SECRET_KEY=NUEVA_CLAVE_SECRETA
   ```

4. **Probar Integración**
   - Realizar un pago de prueba
   - Verificar que se procesa correctamente

---

3. DocuSign

¿Por qué es crítico?
DocuSign maneja documentos legales y contratos. Una clave comprometida permite:
- Acceso a documentos confidenciales
- Firma no autorizada de contratos
- Modificación de documentos

Pasos de Rotación:

1. **Acceder a DocuSign Admin**
   - Ir a: https://admindemo.docusign.com/ (demo) o https://admin.docusign.com/ (producción)
   - Iniciar sesión con credenciales de administrador

2. **Revocar Integración Actual**
   - Navegar a: Settings > Integrations > Apps and Keys
   - Encontrar tu aplicación actual
   - Hacer clic en “Delete” o “Revoke”

3. **Crear Nueva Integración**
   - Hacer clic en “Add App and Integration Key”

- Completar información de la aplicación
- Copiar el nuevo Integration Key

4. **Generar Nuevas Credenciales OAuth**
   - En la aplicación recién creada, generar:
     - Client ID
     - Client Secret
     - RSA Private Key (si usas JWT)
5. **Actualizar Credenciales**

```
bash
  # En tu archivo .env
  DOCUSIGN_INTEGRATION_KEY=NUEVA_INTEGRATION_KEY
  DOCUSIGN_CLIENT_ID=NUEVO_CLIENT_ID
  DOCUSIGN_CLIENT_SECRET=NUEVO_CLIENT_SECRET
  DOCUSIGN_PRIVATE_KEY="-----BEGIN RSA PRIVATE KEY-----\n...\n-----END RSA PRIVATE KEY-----"
```

6. **Probar Integración**
   - Enviar un documento de prueba para firma
   - Verificar que se envía y firma correctamente

---

4. Database (PostgreSQL)

¿Por qué es crítico?
La base de datos contiene TODOS los datos de tu aplicación.

Pasos de Rotación:

1. **Conectar a tu Base de Datos**

```
bash
  psql -h tu-host -U tu-usuario -d tu-database
```

2. **Cambiar Contraseña del Usuario**

```
sql
  ALTER USER tu_usuario WITH PASSWORD 'nueva_contraseña_segura_aqui';
```

3. **O Crear un Nuevo Usuario**

```
sql
  CREATE USER nuevo_usuario WITH PASSWORD 'contraseña_segura';
  GRANT ALL PRIVILEGES ON DATABASE tu_database TO nuevo_usuario;
  GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO nuevo_usuario;
  GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO nuevo_usuario;
```

4. **Actualizar DATABASE_URL**

```
bash
  # En tu archivo .env
  DATABASE_URL="postgresql://nuevo_usuario:contraseña_segura@host:5432/database?schema=public"
```

5. **Probar Conexión**
   ```bash
   cd nextjs_space
   yarn prisma db pull
   ```

6. **Revocar Usuario Antiguo (Opcional pero Recomendado)**
   ```sql
   REVOKE ALL PRIVILEGES ON DATABASE tu_database FROM usuario_viejo;
   DROP USER usuario_viejo;
   ```

---

5. NextAuth Secret

¿Por qué es crítico?
El NextAuth secret se usa para encriptar tokens de sesión. Si está comprometido:
- Un atacante puede falsificar sesiones
- Acceso no autorizado a cuentas de usuario

Pasos de Rotación:

1. **Generar Nuevo Secret**
   ```bash
   openssl rand -base64 32
   ```

2. **Actualizar en .env**
   ```bash
   NEXTAUTH_SECRET="nuevo_secret_generado_aqui"
   ```

3. ⚠️ **IMPORTANTE: Esto Invalidará Todas las Sesiones Actuales**
   - Todos los usuarios tendrán que volver a iniciar sesión
   - Planificar esto durante horas de bajo tráfico si es posible

---

🛡️ Medidas Preventivas Futuras

1. Configurar .gitignore Correctamente

Asegúrate de que tu `.gitignore` incluye:

```
# Environment variables
.env
.env.local
.env.development
.env.production
.env.*.local

# Secrets
secrets/
*.pem
*.key
*.p12
*.pfx

# Config files with credentials
config/*.json
!config/example.json
```

2. Usar Variables de Entorno en Producción

- **Vercel**: Usar la sección “Environment Variables” en el dashboard
- **AWS**: Usar AWS Secrets Manager o Parameter Store
- **Docker**: Pasar variables con `-e` o `--env-file`

3. Auditar Repositorio Regularmente

```
# Instalar herramienta de escaneo
npm install -g git-secrets

# Escanear repositorio
git secrets --scan-history
```

4. Implementar Rotación Automática

Considerar usar:
- **AWS Secrets Manager**: Rotación automática programada
- **HashiCorp Vault**: Gestión centralizada de secretos
- **Azure Key Vault**: Si usas Azure

5. Monitoreo de Seguridad

Configurar alertas para:
- Accesos inusuales a APIs
- Intentos de autenticación fallidos
- Cambios en configuración de seguridad

✅ Checklist de Verificación Post-Rotación

- [ ] Todas las credenciales antiguas han sido revocadas
- [ ] Las nuevas credenciales están en el archivo .env (y NO en Git)
- [ ] La aplicación funciona correctamente con las nuevas credenciales
- [ ] Se ha probado cada integración (Stripe, Redsys, DocuSign)
- [ ] Se ha notificado al equipo del cambio

- [ ] Se ha documentado la fecha de rotación
- [ ] Se ha actualizado la documentación interna
- [ ] Se ha configurado .gitignore para prevenir futuros incidentes
- [ ] Se han revisado los logs para actividad sospechosa
- [ ] Se ha considerado notificar a los usuarios (si aplica)

📞 Contactos de Soporte

Stripe

- Dashboard: https://dashboard.stripe.com
- Soporte: https://support.stripe.com
- Teléfono: Disponible en el dashboard

Redsys

- Soporte: soporte@redsys.es
- Teléfono: +34 91 270 81 00

DocuSign

- Soporte: https://support.docusign.com
- Teléfono: Disponible en el centro de soporte

📚 Referencias

- OWASP Secrets Management (https://cheatsheetseries.owasp.org/cheatsheets/Secrets_Management_Cheat_Sheet.html)
- Stripe Security Best Practices (https://stripe.com/docs/security/guide)
- GitHub Security Best Practices (https://docs.github.com/en/code-security/getting-started/securing-your-repository)

Última actualización: 2024-12-08
Contacto de seguridad: [Tu email de seguridad]