Informe de Laboratorio

Detección y Remediación de Secretos en Repositorios con Gitleaks

Autor: Sebastián Cisneros

Fecha: Septiembre 2025

Ingeniería de Sistemas

Banco de la República - Ipiales

1. Objetivo

Detectar secretos (API keys, contraseñas, tokens) en un repositorio local usando Gitleaks, analizar el riesgo y aplicar remediación.

2. Entorno y herramientas

Sistema: Windows (PowerShell)

Herramientas: Git, Gitleaks v8.28.0, Notepad/VS Code, BFG Repo-Cleaner (opcional).

3. Preparación del repositorio

Se creó un repositorio Git local con archivos que contienen 'secretos' simulados (API keys y contraseñas).

4. Instalación y verificación de Gitleaks

Se descargó gitleaks.exe y se verificó la instalación con el comando:

```
.\gitleaks.exe --version
```

5. Escaneo con Gitleaks

Se ejecutaron distintos modos:

- Historial: analiza commits.

```
.\gitleaks.exe detect --source . --report-format json --report-path report_history.json
```

- Workspace (archivos actuales):

```
.\gitleaks.exe detect --source . --no-git --report-format json --report-path report_workspace.json
```

6. Interpretación del reporte

Los reportes JSON muestran campos como RuleID, Description, File, Secret y Match, indicando dónde y qué se detectó.

7. Remediación

1. git rm archivo.txt

2. Actualización de .gitignore

3. Commit con la eliminación.

Se elimina el secreto de la versión actual, aunque sigue en el historial.

8. Eliminación del historial (BFG)

Con BFG Repo-Cleaner se reemplazan o eliminan secretos de todos los commits pasados, limpiando el historial completo.

9. Verificación final

Se ejecuta nuevamente Gitleaks. El resultado esperado es 'no leaks found', demostrando que la limpieza fue efectiva.

10. Conclusión

Gitleaks es una herramienta esencial para mantener repositorios libres de secretos sensibles. Este laboratorio permitió practicar detección, análisis y remediación de riesgos de seguridad en Git.