CWE-532: Insertion of Sensitive Information into Log File

# Descrição Detalhada

A vulnerabilidade CWE-532 ocorre quando informações sensíveis, como senhas, tokens, dados bancários ou PII (Informações Pessoais Identificáveis), são registradas em arquivos de log. Essa prática pode resultar em vazamentos de dados críticos e violação de regulamentações de privacidade como LGPD, GDPR, HIPAA, etc.

# Exemplo em Java

## Código Vulnerável

import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
public class LoginService {  
 private static final Logger logger = LogManager.getLogger(LoginService.class);  
  
 public boolean authenticate(String username, String password) {  
 logger.info("Tentativa de login: " + username + ", senha: " + password); // ❌ Senha logada!  
 return "admin".equals(username) && "secret".equals(password);  
 }  
}

## Código Corrigido

public boolean authenticate(String username, String password) {  
 logger.info("Tentativa de login com usuário: {}", username); // ✅ Sem log de senha  
 return "admin".equals(username) && "secret".equals(password);  
}

# Exemplo em Python

## Código Vulnerável

import logging  
  
logging.basicConfig(filename='app.log', level=logging.INFO)  
  
def login(user, password):  
 logging.info(f"Usuário: {user}, Senha: {password}") # ❌ log inseguro  
 return user == "admin" and password == "admin123"

## Código Corrigido

def login(user, password):  
 logging.info(f"Tentativa de login com usuário: {user}") # ✅ senha omitida  
 return user == "admin" and password == "admin123"

# Exemplo em C#

## Código Vulnerável

using NLog;  
  
public class AuthService {  
 private static Logger logger = LogManager.GetCurrentClassLogger();  
  
 public bool Login(string username, string password) {  
 logger.Info($"Login: {username}, Senha: {password}"); // ❌ Informação sensível no log  
 return username == "admin" && password == "123456";  
 }  
}

## Código Corrigido

public bool Login(string username, string password) {  
 logger.Info($"Login tentado com usuário: {username}"); // ✅ senha excluída  
 return username == "admin" && password == "123456";  
}

# Boas Práticas de Mitigação

- Nunca registre senhas, tokens, chaves de API ou dados pessoais completos em logs.  
- Use mascaramento parcial, se necessário.  
- Habilite políticas de retenção e criptografia de logs.  
- Faça auditoria e revisão periódica dos sistemas de log.  
- Implemente controles de acesso aos arquivos de log.

# Recomendações de Segurança para CWE-532

## Recomendações Gerais de Segurança de Logs

1. Nunca logar dados sensíveis: Evite registrar senhas, tokens, dados bancários ou PII em logs.  
2. Utilizar mascaramento de dados: Exiba apenas partes de informações sensíveis, como últimos dígitos de cartões.  
3. Filtrar e sanitizar entradas de log: Evite injeção de log com caracteres de controle como \n ou \r.  
4. Separar logs sensíveis por nível ou canal: Utilize níveis e arquivos distintos para logs críticos.  
5. Controle de acesso aos arquivos de log: Restrinja permissões de leitura e escrita aos arquivos de log.  
6. Criptografia e integridade de logs: Armazene logs de forma segura e com validação de integridade.  
7. Auditoria e rotação de logs: Realize auditorias periódicas e implemente rotação automática.  
8. Conformidade legal: Certifique-se de que os logs sigam legislações como LGPD, GDPR ou HIPAA.

## Recomendações por Linguagem

### Java

- Use bibliotecas como Logback ou log4j2 com filtros de sensibilidade.  
- Evite logar objetos diretamente com toString() se contiverem informações sensíveis.  
- Utilize MDC (Mapped Diagnostic Context) para fornecer contexto seguro.

### Python

- Configure logging.Filter para filtrar informações sensíveis dinamicamente.  
- Use frameworks como structlog para maior controle.  
- Cuidado com tracebacks automáticos que podem incluir variáveis sensíveis.

### C#

- Use NLog, Serilog ou log4net com sanitização de logs configurada.  
- Utilize Destructuring Policies para limitar dados expostos em objetos.  
- Desative logs automáticos de frameworks quando incluírem informações sigilosas.