Contents

1 SEC-001: Bypass de Autenticación

1.1 Severidad

Crítica

1.2 Referencia OWASP

A07:2021 - Fallos de Identificación y Autenticación

1.3 Referencias CWE

- CWE-287: Autenticación Impropia
- CWE-306: Falta de Autenticación para Función Crítica

1.4 Descripción

Las vulnerabilidades de bypass de autenticación ocurren cuando un atacante puede acceder a recursos protegidos o realizar acciones privilegiadas sin proporcionar credenciales válidas. En servidores MCP, esta vulnerabilidad permite a usuarios no autorizados invocar funciones protegidas, acceder a datos sensibles o manipular la configuración del servidor sin verificación de identidad adecuada.

Esto es particularmente crítico en implementaciones de MCP porque:

- Los servidores MCP a menudo manejan operaciones sensibles como ejecución de código, acceso al sistema de archivos y transformación de datos
- La autenticación es la primera línea de defensa en la seguridad de las interacciones servidor-cliente
- Las vulnerabilidades de bypass pueden llevar al compromiso completo del servidor MCP y sus recursos conectados

1.5 Mapeo de Cumplimiento

1.5.1 SOC2

- **CC6.1**: Controles de Acceso Lógico y Físico - La organización implementa medidas de seguridad de acceso lógico para proteger contra acceso no autorizado, incluyendo autenticación y autorización.
- **CC6.2**: Control de Acceso - Las identidades de usuario y credenciales se gestionan a través de procedimientos de control de acceso definidos.

1.5.2 HIPAA

- **§164.312(a)(1)**: Identificación de Usuario y Autenticación - Se implementan credenciales de inicio de sesión del sistema de información y mecanismos de control.
- **§164.312(d)**: Cifrado y Descifrado - Se utilizan mecanismos de cifrado apropiados para proteger la información de salud protegida electrónica.

1.5.3 PCI DSS

- **6.2.4**: Los parches de seguridad deben instalarse dentro de un mes de su lanzamiento
- **8.3.1**: La autenticación multifactor debe implementarse para todo acceso no consola
- **8.4.1**: El acceso debe controlarse usando IDs de usuario únicos y mecanismos de autenticación fuerte

1.6 Ejemplo de Código Vulnerable

```
// □ INSEGURO: Sin middleware de autenticación
const express = require('express');
const app = express();

// Acceso direto a funciones sensibles sin ninguna verificación de autenticación
app.post('/api/execute-tool', (req, res) => {
  const { toolName, params } = req.body;

  // Este endpoint está completamente expuesto - cualquiera puede llamarlo
  const result = executeTool(toolName, params);

  res.json({ success: true, result });
});

app.delete('/api/config', (req, res) => {
  // Sin verificación de que el usuario tiene permiso para eliminar la configuraci
  deleteServerConfig();
  res.json({ success: true, message: 'Configuration deleted' });
});
```

```
app.listen(3000);
```

1.7 Implementación Segura

```
// ✅ SEGURO: Middleware de autenticación apropiado con tokens JWT
const express = require('express');
const jwt = require('jsonwebtoken');
const app = express();

const SECRET_KEY = process.env.JWT_SECRET;

// Middleware de autenticación
const authenticateToken = (req, res, next) => {
  const authHeader = req.headers['authorization'];
  const token = authHeader && authHeader.split(' ')[1]; // Bearer TOKEN

  if (!token) {
    return res.status(401).json({ error: 'Token de acceso requerido' });
  }

  jwt.verify(token, SECRET_KEY, (err, user) => {
    if (err) {
      return res.status(403).json({ error: 'Token inválido o expirado' });
    }
    req.user = user;
    next();
  });
};

// Middleware de autorización basado en roles
const authorizeRole = (requiredRole) => {
  return (req, res, next) => {
    if (!req.user || req.user.role !== requiredRole) {
      return res.status(403).json({ error: 'Permisos insuficientes' });
    }
    next();
  };
};

// Endpoint protegido con autenticación
app.post('/api/execute-tool', authenticateToken, (req, res) => {
  const { toolName, params } = req.body;

  // La identidad del usuario se verifica a través del token JWT
  console.log(`Ejecución de herramienta solicitada por usuario: ${req.user.id}`);
```

```
  const result = executeTool(toolName, params);
  res.json({ success: true, result });
});

// Endpoint protegido con autorización basada en roles
app.delete('/api/config',
  authenticateToken,
  authorizeRole('admin'),
  (req, res) => {
    // Solo usuarios administradores autenticados pueden eliminar la configuración
    console.log(`Configuración eliminada por admin: ${req.user.id}`);
    deleteServerConfig();
    res.json({ success: true, message: 'Configuración eliminada' });
  }
);

app.listen(3000);
```

1.8 Pasos de Remediación

1. **Implementar Autenticación Basada en Tokens**: Utiliza tokens JWT (JSON Web Tokens) estándar de la industria u OAuth 2.0 para todos los endpoints de API. Asegúrate de que los tokens se generen solo después de la validación exitosa de credenciales e incluyan tiempos de expiración (típicamente 15-60 minutos para tokens de acceso).
2. **Aplicar Middleware de Autenticación**: Crea middleware que valide tokens de autenticación en cada endpoint protegido. Este middleware debe verificar la validez del token, la expiración y la firma antes de permitir el acceso a funciones sensibles.
3. **Implementar Control de Acceso Basado en Roles (RBAC)**: Define roles de usuario con permisos específicos y valida que los usuarios autenticados tengan el rol/permisos requeridos para cada operación. Documenta qué roles pueden acceder a qué endpoints.
4. **Usar HTTPS y Almacenamiento Seguro de Tokens**: Siempre transmite credenciales de autenticación sobre HTTPS para prevenir interceptación. Almacena tokens de forma segura (cifrados) e implementa políticas de rotación de tokens apropiadas. Nunca almacenes credenciales sensibles en código o archivos de configuración.

1.9 Referencias

- OWASP Authentication Cheat Sheet
- NIST SP 800-63B: Autenticación y Gestión del Ciclo de Vida
- CWE-287: Autenticación Impropia
- JWT.io: Introducción a JSON Web Tokens
- PCI DSS Requisito 8: Identificación y Autenticación

Última Actualización: Enero 2026
Estado: Publicado
Idioma: Español