🔒 SECURITY ASSESSMENT REPORT

Model Context Protocol Server  
http://test.example.com:3000

|  |  |
| --- | --- |
| **Report Date:** | October 16, 2025 |
| **Target Server:** | http://test.example.com:3000 |
| **Server Name:** | Test Server |
| **Assessment Type:** | Comprehensive Security Scan |
| **Status:** | CONFIDENTIAL |

# EXECUTIVE SUMMARY

This report presents the findings from a comprehensive security assessment of the Model Context Protocol (MCP) server deployed at http://test.example.com:3000. The assessment identified 7 security issues that require attention to ensure the confidentiality, integrity, and availability of the system.

**OVERALL RISK SCORE: 33/100  
LOW RISK**

## Vulnerability Breakdown

|  |  |  |  |
| --- | --- | --- | --- |
| **SEVERITY** | **COUNT** | **RISK LEVEL** | **PRIORITY** |
| Critical | 2 | 🔴 Immediate | P0 - Fix Now |
| High | 2 | 🟠 Urgent | P1 - This Week |
| Medium | 1 | 🟡 Important | P2 - This Month |
| Low | 1 | 🔵 Minor | P3 - Backlog |
| Info | 1 | 🟢 FYI | P4 - Optional |

# DETAILED FINDINGS

## 1. Missing Authentication

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | MCP-AUTH-001 | **Severity:** | CRITICAL |
| **Category:** | Authentication | **CWE:** | CWE-306 |
| **CVSS:** | 9.8/10.0 |  |  |

**Description:**

The MCP server at http://test.example.com:3000 does not require authentication. Any client can connect and access all available tools and resources.

**Evidence:**

* • Server URL: http://test.example.com:3000
* • Successfully connected without credentials
* • Available tools: 2
* • Available resources: 1

**Remediation:**

Implement authentication mechanism such as:

* API keys
* OAuth 2.0
* Mutual TLS (mTLS)
* JWT tokens

## 2. Dangerous Tools Exposed Without Authorization

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | MCP-AUTHZ-001 | **Severity:** | CRITICAL |
| **Category:** | Authorization | **CWE:** | CWE-285 |
| **CVSS:** | 9.1/10.0 |  |  |

**Description:**

The MCP server exposes 2 potentially dangerous tools without proper authorization controls. Combined with missing authentication, these tools can be abused for system compromise.

**Evidence:**

* • Dangerous tools found: read\_file, execute
* • Total tools exposed: 2
* • Authentication required: False

**Remediation:**

Implement proper authorization:

* Role-based access control (RBAC)
* Principle of least privilege
* Input validation for all tools
* Audit logging for tool usage

## 3. Unencrypted Connection

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | MCP-CRYPTO-001 | **Severity:** | HIGH |
| **Category:** | Encryption | **CWE:** | CWE-319 |
| **CVSS:** | 7.5/10.0 |  |  |

**Description:**

The MCP server at http://test.example.com:3000 does not use TLS/SSL encryption. All communication is transmitted in plaintext, allowing attackers to intercept sensitive data including credentials and API responses.

**Evidence:**

* • Server URL: http://test.example.com:3000
* • Protocol: HTTP (unencrypted)
* • Traffic can be intercepted

**Remediation:**

Enable TLS/SSL encryption:

* Use HTTPS instead of HTTP
* Install valid SSL certificate
* Configure TLS 1.2 or higher
* Disable weak cipher suites

## 4. Command Execution Tools Exposed

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | MCP-INJ-004 | **Severity:** | HIGH |
| **Category:** | Injection | **CWE:** | CWE-78 |
| **CVSS:** | 8.1/10.0 |  |  |

**Description:**

The MCP server at http://test.example.com:3000 exposes command execution tools (execute). These are high-risk tools that could allow command injection if not properly secured.

**Evidence:**

* • Command execution tools: execute
* • High-risk functionality exposed
* • Potential for system compromise

**Remediation:**

Secure command execution tools:

* Implement strict input validation
* Use allowlists for permitted commands
* Never execute shell commands with user input
* Consider removing or restricting these tools
* Require authentication and authorization
* Log all command execution attempts

## 5. File Access Tools With Potential Path Traversal Risk

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | MCP-INJ-006 | **Severity:** | MEDIUM |
| **Category:** | Injection | **CWE:** | CWE-22 |
| **CVSS:** | N/A |  |  |

**Description:**

The MCP server at http://test.example.com:3000 exposes file access tools (read\_file) that may be vulnerable to path traversal if input validation is insufficient.

**Evidence:**

* • File access tools: read\_file
* • Potential path traversal risk
* • Input validation status unknown

**Remediation:**

Secure file access tools:

* Implement strict path validation
* Use allowlists for permitted directories
* Canonicalize paths before access
* Never trust user-supplied file paths
* Restrict file system access scope

## 6. Default Port Configuration

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | MCP-CONFIG-001 | **Severity:** | LOW |
| **Category:** | Configuration | **CWE:** | N/A |
| **CVSS:** | N/A |  |  |

**Description:**

The MCP server is running on a default port (3000). This makes it easier for attackers to discover and target the server.

**Evidence:**

* • Current port: 3000
* • Default port detected

**Remediation:**

Change to a non-standard port:

* Use a random high port (>10000)
* Update firewall rules accordingly
* Document the port change

## 7. Version Information Disclosure

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | MCP-INFO-001 | **Severity:** | INFO |
| **Category:** | Information Disclosure | **CWE:** | N/A |
| **CVSS:** | N/A |  |  |

**Description:**

The server discloses its version (1.0.0), which can help attackers identify known vulnerabilities in that specific version.

**Evidence:**

* • Disclosed version: 1.0.0

**Remediation:**

Minimize information disclosure:

* Remove or obfuscate version headers
* Use generic error messages
* Keep software updated regardless

# RECOMMENDATIONS

**Immediate Actions (Priority 0):**

* • Address all CRITICAL vulnerabilities within 24 hours
* • Implement temporary mitigations if permanent fixes require time
* • Notify security team and stakeholders

**Short-term Actions (1-2 weeks):**

* • Resolve all HIGH severity issues
* • Begin addressing MEDIUM severity vulnerabilities
* • Implement monitoring and alerting

**Long-term Actions (1-3 months):**

* • Address remaining MEDIUM and LOW severity issues
* • Implement security best practices
* • Schedule regular security assessments
* • Provide security training to development team

**Disclaimer:** This report is provided for informational purposes only. The findings represent potential security issues identified through automated and manual testing. Manual verification is recommended before taking remediation actions.

Report Generated by: MCP Security Scanner v0.2.0  
Generated: 2025-10-16 21:27:52