

BurtTheCoder / mcp-dnstwist Public

<> Code Issues 3 Pull requests 4 Actions Projects Security and quality

1 Branch 4 Tags Go to file Go to file <> Code ...

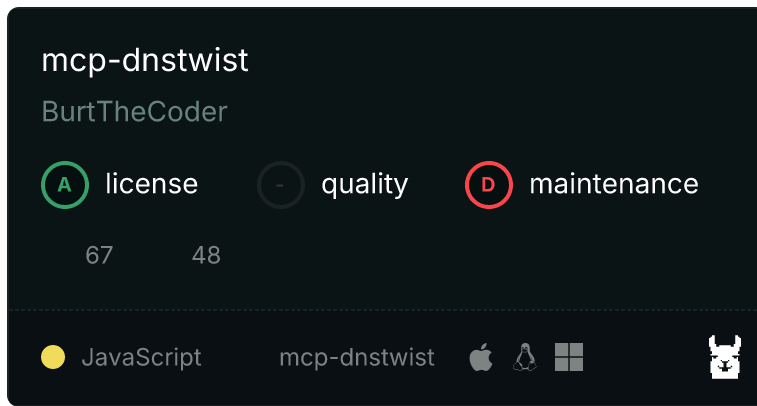
actions-user Bump version to 1.0.4 [skip ci]		af62d09 · last year
📁 .github/workflows	Initial code	2 years ago
📁 src	Initial code	2 years ago
📄 .gitignore	Initial code	2 years ago
📄 Dockerfile	Add Dockerfile	last year
📄 LICENSE	Initial code	2 years ago
📄 README.md	Update README.md	2 years ago
📄 package-lock.json	Bump version to 1.0.4 [skip ci]	last year
📄 package.json	Bump version to 1.0.4 [skip ci]	last year
📄 smithery.yaml	Add Smithery configuration	last year
📄 tsconfig.json	Initial code	2 years ago

README MIT license

# DNStwist MCP Server

📄 Calls ▲ 0

A Model Context Protocol (MCP) server for [dnstwist](#), a powerful DNS fuzzing tool that helps detect typosquatting, phishing, and corporate espionage. This server provides tools for analyzing domain permutations and identifying potentially malicious domains. It is designed to integrate seamlessly with MCP-compatible applications like [Claude Desktop](#).



## Warning

This tool is designed for legitimate security research purposes. Please:

- Only analyze domains you own or have permission to test
- Respect rate limits and DNS server policies
- Use responsibly and ethically
- Be aware that some DNS servers may rate-limit or block automated queries
- Consider the impact on DNS infrastructure when running large scans

## Requirements

- Node.js (v18 or later)
- Docker
- macOS, Linux, or Windows with Docker Desktop installed

## Quick Start

### Installing via Smithery

To install DNStwist for Claude Desktop automatically via [Smithery](#):

```
npx -y @smithery/cli install @burtthecoder/mcp-dnstwist --client claude
```



### Installing Manually

1. Install Docker:
  - macOS: Install [Docker Desktop](#)

- o Linux: Follow the [Docker Engine installation guide](#)

2. Install the server globally via npm:

```
npm install -g mcp-dnstwist
```



3. Add to your Claude Desktop configuration file:

```
{  
  "mcpServers": {  
    "dnstwist": {  
      "command": "mcp-dnstwist"  
    }  
  }  
}
```



Configuration file location:

- macOS: `~/Library/Application Support/Claude/claude_desktop_config.json`
- Windows: `%APPDATA%\Claude\claude_desktop_config.json`

4. Restart Claude Desktop

## Alternative Setup (From Source)

If you prefer to run from source or need to modify the code:

1. Clone and build:

```
git clone <repository_url>  
cd mcp-dnstwist  
npm install  
npm run build
```



2. Add to your Claude Desktop configuration:

```
{  
  "mcpServers": {  
    "dnstwist": {  
      "command": "node",  
      "args": ["/absolute/path/to/mcp-dnstwist/build/index.js"]  
    }  
  }  
}
```



```
}  
}
```

## Features

---

- **Domain Fuzzing:** Generate domain permutations using various algorithms
- **Registration Check:** Verify if permuted domains are registered
- **DNS Analysis:** Check A, AAAA, MX, and NS records
- **Web Presence:** Capture HTTP banner information
- **WHOIS Data:** Retrieve registration dates and registrar information
- **Phishing Detection:** Generate fuzzy hashes of web pages
- **Configurable:** Custom DNS servers and parallel processing
- **Multiple Formats:** Support for json, csv, and list output formats

## Tools

---

### Domain Fuzzing Tool

- Name: `fuzz_domain`
- Description: Generate and analyze domain permutations to detect potential typosquatting, phishing, and brand impersonation
- Parameters:
  - `domain` (required): Domain name to analyze (e.g., example.com)
  - `nameservers` (optional, default: "1.1.1.1"): Comma-separated list of DNS servers
  - `threads` (optional, default: 50): Number of threads for parallel processing
  - `format` (optional, default: "json"): Output format (json, csv, list)
  - `registered_only` (optional, default: true): Show only registered domains
  - `mxcheck` (optional, default: true): Check for MX records
  - `ssdeep` (optional, default: false): Generate fuzzy hashes of web pages
  - `banners` (optional, default: true): Capture HTTP banner information

Example:

```
{  
  "domain": "example.com",  
  "nameservers": "1.1.1.1,8.8.8.8",  
  "threads": 50,  
  "format": "json",  
}
```



```
"registered_only": true,  
"mxcheck": true,  
"banners": true  
}
```

## Troubleshooting

---

### Docker Issues

1. Verify Docker is installed and running:

```
docker --version  
docker ps
```



2. Check Docker permissions:

- Ensure your user has permissions to run Docker commands
- On Linux, add your user to the docker group: `sudo usermod -aG docker $USER`

### Common Issues

1. DNS resolution problems:

- Verify DNS servers are accessible
- Try alternative DNS servers (e.g., 8.8.8.8)
- Check for rate limiting or blocking

2. Performance issues:

- Adjust thread count based on system capabilities
- Consider network bandwidth and latency
- Monitor DNS server response times

3. After fixing any issues:

- Save the configuration file
- Restart Claude Desktop

### Error Messages

---

- "Docker is not installed or not running": Install Docker and start the Docker daemon

- "Failed to parse dnstwist output": Check if the domain is valid and the format is correct
- "Error executing dnstwist": Check Docker logs and ensure proper permissions
- "DNS server not responding": Verify DNS server accessibility and try alternative servers

## Contributing

---

1. Fork the repository
2. Create a feature branch ( `git checkout -b feature/amazing-feature` )
3. Commit your changes ( `git commit -m 'Add amazing feature'` )
4. Push to the branch ( `git push origin feature/amazing-feature` )
5. Open a Pull Request

## License

---

### Releases

 4 tags

---

### Packages

No packages published

---

### Contributors 4



**BurtTheCoder** Burt



**calclavia** Henry Mao



**actions-user**



**punkpeye** Frank Fiegel

---

### Languages



