Security Policy

Supported Versions

We actively support the following versions with security updates:

Version	Supported
0.1.x	:white_check_mark:
< 0.1	:x:

Security Features

Automated Security Scanning

- **CodeQL Analysis**: Automated vulnerability scanning on every push and PR
- **OSSF Scorecard**: Weekly supply chain security assessment
- **Dependency Scanning**: Automated via Dependabot with security updates
- **SBOM Generation**: Software Bill of Materials with attestations

Supply Chain Security

- **Branch Protection**: Main branch requires PR reviews and status checks
- **Signed Commits**: Encouraged for all contributions
- **Dependency Pinning**: All dependencies are pinned to specific versions
- **Provenance**: SLSA provenance generation for releases

Security Best Practices

- **Least Privilege**: All workflows use minimal required permissions
- **Secrets Management**: No hardcoded secrets in repository
- **Input Validation**: All user inputs are validated and sanitized
- **Error Handling**: Secure error handling prevents information disclosure

Reporting a Vulnerability

We take security vulnerabilities seriously. If you discover a security issue, please follow these steps:

1. DO NOT create a public GitHub issue

2. Report privately via GitHub Security Advisories

1. Go to the Security tab (https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen/security)
2. Click "Report a vulnerability"
3. Fill out the security advisory form with:
   - Detailed description of the vulnerability
   - Steps to reproduce

  - Potential impact assessment
  - Suggested fix (if known)

3. Alternative reporting methods

If GitHub Security Advisories are not available, you can:

- Email: Create an issue (https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen/issues/new) with title "SECURITY: [Brief Description]" and mark it as confidential
- Include your contact information for follow-up

4. What to expect

- **Acknowledgment**: Within 48 hours
- **Initial Assessment**: Within 5 business days
- **Regular Updates**: Every 5 business days until resolution
- **Resolution Timeline**: Varies by severity
- Critical: 1-7 days
- High: 7-30 days
- Medium: 30-90 days
- Low: 90+ days

5. Disclosure Policy

- We follow coordinated disclosure principles
- Security fixes will be released before public disclosure
- We will credit reporters (unless they prefer to remain anonymous)
- Public disclosure typically occurs 90 days after fix release

Security Considerations for Users

Installation Security

```
# Verify package integrity when installing from PyPI
pip install utilityfog-fractal-tree --require-hashes

# Or install from verified source
git clone https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen.git
cd UtilityFog-Fractal-TreeOpen
# Verify commit signatures if available
git verify-commit HEAD
pip install -e .
```

Runtime Security

- **Network Security**: All network communications should use TLS
- **Input Validation**: Validate all configuration files and user inputs
- **Logging**: Avoid logging sensitive information
- **File Permissions**: Ensure proper file permissions for configuration files

Configuration Security

```python
# Example secure configuration
config = {
    "telemetry": {
        "export_sensitive_data": False,
        "anonymize_user_data": True,
        "secure_transport": True
    },
    "logging": {
        "log_level": "INFO",  # Avoid DEBUG in production
        "sanitize_logs": True
    }
}
```

Security Architecture

Threat Model

Our security model addresses:

1. **Supply Chain Attacks**: Dependency verification and SBOM generation
2. **Code Injection**: Input validation and sanitization
3. **Data Exfiltration**: Secure data handling and minimal data collection
4. **Privilege Escalation**: Least privilege principles
5. **Denial of Service**: Rate limiting and resource management

Security Controls

Control Type	Implementation
Preventive	Branch protection, code review, input validation
Detective	CodeQL scanning, dependency monitoring, logging
Corrective	Automated patching, incident response procedures
Recovery	Backup procedures, rollback capabilities

Compliance and Standards

- **NIST Cybersecurity Framework**: Aligned with core functions
- **OWASP Top 10**: Mitigations implemented for common vulnerabilities
- **SLSA**: Supply chain security framework compliance
- **OpenSSF**: Best practices implementation

Security Resources

- OWASP Secure Coding Practices (https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/)
- NIST Cybersecurity Framework (https://www.nist.gov/cyberframework)
- OpenSSF Best Practices (https://bestpractices.coreinfrastructure.org/)
- SLSA Framework (https://slsa.dev/)

Contact

For security-related questions or concerns:

- **Security Team**: Use GitHub Security Advisories
- **General Security Questions**: Create a GitHub Discussion
- **Community**: Join our security-focused discussions

Last Updated: September 2025
Next Review: December 2025