OmniDork: User Manual

Installation

Prerequisites

- Rust toolchain (1.60 or newer)
- PostgreSQL database (for storing results)
- OpenSSL development libraries
- pkg-config

Installation Steps

1. Clone the repository:

   bash

   ```bash
   git clone https://github.com/your-username/omnidork.git
   cd omnidork
   ```

2. Create a `.env` file with your database configuration:

   ```
   DATABASE_URL=postgres://username:password@localhost/omnidork
   ```

3. Create the database:

   bash

   ```bash
   psql -c "CREATE DATABASE omnidork;"
   ```

4. Install and build the project:

   bash

   ```bash
   cargo build --release
   ```

Getting Started

After installation, run OmniDork from the command line:

bash

```bash
./target/release/omnidork
```

This will display the main menu with five operating modes:

1. OSINT and Vulnerability Scanning
2. Quantum Resonant Search

3. Proxy Scanning
4. Open Redirect Vulnerability Scan
5. Full Integrated Scan

Mode 1: OSINT and Vulnerability Scanning

This mode focuses on discovering security vulnerabilities in a target domain.

Usage

1. Select option 1 from the main menu
2. Enter a target domain (e.g., example.com)
3. OmniDork will automatically:
    - Discover subdomains
    - Execute Google dorks
    - Query security services (Shodan, URLScan)
    - Analyze JavaScript files
    - Check for cloud storage exposures
    - Generate a vulnerability report

Example

```
[1/9] Performing domain reconnaissance...
Found 12 subdomains

[2/9] Executing dorks against target and subdomains...
Got 47 results from dork queries

[3/9] Querying specialized security services...
Retrieved data from Shodan, URLScan.io, and DNS lookups

[4/9] Extracting and analyzing JavaScript files...
Analyzed 8 JavaScript files

[5/9] Checking for cloud storage resources...
Found 3 cloud storage resources

[6/9] Analyzing findings...
Analysis complete with 17 findings

[7/9] Scanning for usable proxies...
Found 22 working proxies

[8/9] Generating visualizations...
Generated 3 visualizations

[9/9] Matching findings to bug bounty programs...
Found 5 potential bug bounty matches
```

Interpreting Results

The tool saves all results in the `data/findings/` directory with a comprehensive JSON report, which includes:

- Vulnerability details and severity
- URLs and descriptions
- Potential bug bounty matches
- Evidence like screenshots and snippets

Mode 2: Quantum Resonant Search

This mode provides an advanced search engine that uses quantum-inspired algorithms to find relevant content.

Usage

1. Select option 2 from the main menu

2. Configure settings:
    - Enable quantum-inspired scoring (recommended)
    - Enable persistence theory scoring (recommended)
    - Set fragility parameter (0.1-1.0)
    - Set entropy weight (0.1-1.0)
3. Choose your data source:
    - Default seed URLs
    - Custom URL list
    - Single domain to crawl
    - Existing index
4. Configure crawling parameters:
    - Number of pages to crawl
    - Maximum depth
    - Number of concurrent workers
5. Enter search queries when crawling is complete

Example Search

```
Enter your resonant query (or type 'quit' to exit):
> quantum cryptography applications

Searching for resonant matches...

Top Resonant Matches:
[1] Quantum Cryptography: Practical Applications and Future Directions
    URL:            https://example.com/quantum-crypto.html
    Resonance:      0.8754
    Δ Entropy:      0.1123
    Standard Score: 0.7631
    Quantum Score:  0.8912
    Persist. Score: 0.7655
    Combined Score: 0.7957
    Preview:        Quantum cryptography offers several practical applications in secure communications...
```

Special Commands

During the search phase, you can use these special commands:

- `export`: Export the current index to CSV

- `checkpoint`: Save a checkpoint of the current state
- `compress`: Compress all documents to save memory
- `quit`: Exit the search mode

Mode 3: Proxy Scanner

This mode discovers and validates anonymous proxies from multiple sources.

Usage

1. Select option 3 from the main menu
2. Configure scanner settings:
   - Maximum concurrent connections
   - Number of validation rounds
   - Connection timeout
   - Anonymity checking
3. Wait for the scanner to find working proxies
4. Optionally run speed tests on discovered proxies

Proxy Types

OmniDork classifies proxies into three anonymity levels:

- **Elite**: Your real IP is completely hidden; the proxy doesn't reveal itself
- **Anonymous**: The proxy identifies itself as a proxy but doesn't reveal your real IP
- **Transparent**: The proxy reveals your real IP address in headers

Example Output

```
Found working proxy: 123.45.67.89:8080 (256.32ms, France)
Found working proxy: 98.76.54.32:3128 (312.45ms, Germany)
Found working proxy: 45.67.89.12:8888 (427.18ms, Netherlands)

Proxy scan complete!
Found 22 working proxies
Saved working proxies to data/proxies/working_proxies_1684971234.txt

Speed test results:
1. 98.76.54.32:3128 - 142.35ms
2. 123.45.67.89:8080 - 183.27ms
3. 11.22.33.44:80 - 198.56ms
```

Mode 4: Open Redirect Vulnerability Scan

This mode focuses specifically on finding open redirect vulnerabilities in a list of URLs.

Usage

1. Select option 4 from the main menu
2. Enter the path to a file containing URLs to scan
3. Enter the payload for redirect testing (default: http://evil.com)
4. Wait for the scan to complete

Example Output

```
Loaded 150 URLs from targets.txt
Starting open redirect scan with payload: http://evil.com

Open Redirect Found: https://example.com/redirect?url=http://evil.com
Open Redirect Found: https://another-site.org/go?to=http://evil.com
Redirect to different location for https://test.com/redir?url=http://evil.com:
https://test.com/error

Scan complete!
Found 7 vulnerable URLs
Saved vulnerable URLs to data/findings/open_redirect_vulns_1684972345.txt
```

Mode 5: Full Integrated Scan

This mode combines all previous capabilities for a comprehensive security analysis.

Usage

1. Select option 5 from the main menu
2. Enter the target domain for OSINT scanning
3. Wait for all components to complete their scans
4. View the integrated report

What It Does

The full integrated scan:

1. Runs OSINT to discover the attack surface
2. Applies quantum resonant crawling to index discovered content
3. Finds proxies that can be used for anonymous scanning
4. Analyzes everything together in a comprehensive way

5. Matches findings to bug bounty programs
6. Generates a complete markdown report

Advanced Features

Customizing Dork Patterns

You can add custom dork patterns by editing `src/dork_engine.rs`. Look for the `dork_categories` HashMap and add your patterns following the existing format.

Using Found Proxies

To use the discovered proxies with other tools:

1. Run a proxy scan (Mode 3)
2. Find the saved proxy list in `data/proxies/`
3. Configure your other tools to use these proxies

Database Integration

OmniDork stores all findings in a PostgreSQL database for easy querying and historical analysis. Example queries:

```sql
-- Get all high severity findings
SELECT * FROM findings WHERE severity = 'High';

-- Count findings by type
SELECT finding_type, COUNT(*) FROM findings GROUP BY finding_type;

-- Find domains with the most vulnerabilities
SELECT target_id, COUNT(*) FROM findings GROUP BY target_id ORDER BY COUNT(*) DESC;
```

Troubleshooting

Common Issues

1. **Connection errors during dorking**:
   - This is normal and may be due to rate limiting
   - The tool will continue with other dorks automatically
2. **Slow crawling performance**:
   - Reduce the maximum crawl depth
   - Decrease the number of concurrent workers

3. **High memory usage**:
    - Use the `compress` command during search to free memory
    - Reduce the number of pages to crawl
4. **No proxies found**:
    - Increase the connection timeout
    - Try adding more proxy sources in `proxy_scanner.rs`

Contributing

We welcome contributions to OmniDork! Here's how to get started:

1. Fork the repository
2. Create a new branch for your feature
3. Add your changes
4. Submit a pull request

Please follow the Rust style guidelines and include tests for new features.

License

OmniDork is licensed under the MIT License. See the LICENSE file for details.

Contact

For questions, issues, or contributions, please contact us at:

- GitHub Issues: https://github.com/your-username/omnidork/issues
- Email: your.email@example.com

Thank you for using OmniDork! Happy hacking (ethically, of course)!