Enhanced Web Security Scanner - User Guide

Overview

The Enhanced Web Security Scanner is a comprehensive, non-intrusive vulnerability assessment tool designed for educational purposes and authorized penetration testing. It performs automated security checks against web applications and generates detailed reports.

Features

- **Passive Reconnaissance**: Crawls websites to discover pages, forms, and endpoints
- **Vulnerability Detection**:
  - Cross-Site Scripting (XSS) - Reflected
  - SQL Injection (Error-based, Time-based, Boolean-based)
  - Directory Traversal
  - HTTP Parameter Pollution
  - Missing Security Headers
  - Insecure Cookie Configuration
  - CSRF Protection Analysis
- **Comprehensive Reporting**: Generates Markdown, HTML, and JSON reports
- **Technology Stack Detection**: Identifies web servers, frameworks, and technologies
- **Rate Limiting**: Configurable request throttling to avoid overwhelming targets

Installation

Prerequisites

- Python 3.7 or higher
- pip package manager

Required Dependencies

```
pip install requests beautifulsoup4 lxml markdown2
```

Installation Steps

1. **Clone or Download** the scanner files
2. **Install dependencies**:

   ```
   pip install -r requirements.txt
   ```

3. **Verify installation**:

   ```
   python3 simple_scanner.py --help
   ```

requirements.txt

```
requests>=2.28.0
beautifulsoup4>=4.11.0
lxml>=4.9.0
markdown2>=2.4.0
```

Usage

Basic Usage

```
python3 simple_scanner.py -u http://target-url.com
```

Command Line Options

```
-u, --url         Target URL (required)
-m, --max         Maximum pages to crawl (default: 30)
--output          Output file path (default: reports/enhanced_scan_report.md)
--json-only       Generate only JSON report
--no-html         Skip HTML report generation
--rate-limit      Seconds between requests (default: 0.8)
```

Examples

Basic scan of DVWA:

```
python3 simple_scanner.py -u http://localhost/dvwa
```

Comprehensive scan with increased page limit:

```
python3 simple_scanner.py -u http://testphp.vulnweb.com -m 50
```

Fast scan with reduced rate limiting:

```
python3 simple_scanner.py -u http://localhost:3000 --rate-limit 0.3
```

Generate only JSON report:

```
python3 simple_scanner.py -u http://target.com --json-only
```

Configuration

Rate Limiting

The scanner includes built-in rate limiting to avoid overwhelming target servers:

- Default: 0.8 seconds between requests
- Adjustable via `-rate-limit` parameter
- Recommended: 0.5-1.0 seconds for local testing, 1.0+ for external targets

Scan Depth

- Default maximum pages: 30
- Adjustable via `m` parameter
- Consider target size and scan time when adjusting

Output Formats

1. **Markdown Report**: Human-readable format with detailed findings
2. **HTML Report**: Styled web-viewable format
3. **JSON Report**: Machine-readable format for integration

Understanding Reports

Risk Levels

- 🔴 **High Risk**: Critical vulnerabilities requiring immediate attention
  - XSS vulnerabilities
  - SQL injection flaws
  - Directory traversal
- 🟡 **Medium Risk**: Important security improvements
  - Missing security headers
  - Insecure cookies
  - Parameter pollution
- 🟢 **Low Risk**: Information disclosure and minor issues

Report Sections

1. **Executive Summary**: High-level overview of findings
2. **Technology Stack**: Detected server and framework information
3. **High Risk Vulnerabilities**: Critical security issues
4. **Medium Risk Issues**: Important security hardening items

5. **Information Disclosure**: Accessible files and directories
6. **Forms Analysis**: Detailed form enumeration
7. **CSRF Protection**: Token analysis
8. **Recommendations**: Prioritized remediation guidance

Troubleshooting

Common Issues

Connection Errors:

- Verify target URL is accessible
- Check network connectivity
- Ensure target accepts HTTP requests

SSL/TLS Errors:

- Add certificates to Python's certificate store
- Use HTTP instead of HTTPS for local testing

Permission Errors:

- Ensure write permissions for reports directory
- Run with appropriate user privileges

Memory Issues with Large Sites:

- Reduce max pages limit (-m parameter)
- Increase rate limiting to reduce concurrent load

Error Messages

- `fetch error`: Network connectivity issues
- `Invalid URL`: Malformed target URL
- `Scan interrupted`: User cancelled (Ctrl+C)

Advanced Usage

Custom Payload Testing

The scanner includes multiple payload sets for comprehensive testing:

- **XSS Payloads**: 8 different context-aware payloads
- **SQL Injection**: 8 payloads covering different injection types
- **Directory Traversal**: 4 payloads for different OS types

Output File Structure

```
reports/
├── enhanced_scan_report.md     # Markdown report
├── enhanced_scan_report.html   # HTML report
└── scan_results.json           # JSON data
```

Technical Details

Scan Process

1. **URL Normalization**: Ensures consistent URL format
2. **Initial Reconnaissance**: Fetches base page and analyzes headers
3. **Link Discovery**: Crawls discovered pages up to specified limit
4. **Form Analysis**: Identifies and catalogs all forms
5. **Vulnerability Testing**: Systematic testing of identified attack vectors
6. **Path Discovery**: Tests for common sensitive files/directories
7. **Report Generation**: Creates comprehensive reports in multiple formats

Detection Methods

- **XSS**: Reflection-based detection with unique markers
- **SQL Injection**: Error pattern matching, timing analysis, and boolean-based detection
- **Directory Traversal**: Content pattern matching for system files

- **Security Headers**: Comprehensive header analysis with configuration validation