

 CASABA SECURITY, LLC

08

**Fall**

Watcher Functional Specification

Chris Weber

This document is a blueprint for how the Watcher Security Auditor application will look and work. It details what the finished product will do, how a user will interact with it, and what it will look like.

|  |  |  |  |
| --- | --- | --- | --- |
| **Watcher — Functional Specification** | | | |
| Program Manager | Chris Weber | Release | 1.3 |
| Development Lead | Rob Mooney, Samuel Bucholtz | Status | In review |
| Test Lead | Chris Weber | Version | See change history |

This document is for internal Casaba Security use only. Any distribution to external parties must be cleared by the author requires a scrubbing of the document for external consumption. This document does not represent a commitment by Casaba Security and implementation is subject to change.

Table of Contents

[Change History 3](#_Toc230591265)

[Overview 3](#_Toc230591266)

[Value Proposition 3](#_Toc230591267)

[Business Justification 3](#_Toc230591268)

[Goals 3](#_Toc230591269)

[Scope 3](#_Toc230591270)

[Glossary & Bibliography 3](#_Toc230591271)

[Glossary 3](#_Toc230591272)

[Bibliography 4](#_Toc230591273)

[Scenarios 4](#_Toc230591274)

[Scenario 1: Developer or tester wants an assessment of their application between release cycles. 4](#_Toc230591275)

[Scenario 2: Pen-tester wants to find hot-spots for vulnerabilities 4](#_Toc230591276)

[Scenario 3: Auditor wants to check a Web-apps compliance against industry standards such as PCI, OWASP. 4](#_Toc230591277)

[Requirements 4](#_Toc230591278)

[Supported 4](#_Toc230591279)

[Not Supported 5](#_Toc230591280)

[Assumptions 5](#_Toc230591281)

[Details 5](#_Toc230591282)

[Functionality 5](#_Toc230591283)

[User Interface 5](#_Toc230591284)

[Customization, Globalization & Accessibility 6](#_Toc230591285)

[Platform 6](#_Toc230591286)

[Performance 6](#_Toc230591287)

[Frequently Asked Questions 6](#_Toc230591288)

[Future Enhancements 6](#_Toc230591289)

[Related Documents 6](#_Toc230591290)

[Open Issues 6](#_Toc230591291)

# Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Version | Date | Comment |
| Chris Weber | 1.0 | 06/05/2008 | Initial straw man created. |
| Chris Weber | 1.1 | 05/20/2009 | Updates for version 1.3 release |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Overview

Watcher is a passive security analysis tool for HTTP based Web-apps. It’s a plugin for Fiddler.

## Value Proposition

For Casaba Security:

* Demonstrate leadership in Web-application security testing.

For Users:

* Easy validation that applications meet security requirements or not.

## Business Justification

See Value Proposition, above.

## Goals

* Provide a core engine for inspecting HTTP request/responses
* Provide a set of checks to inspect HTTP traffic
* Provide a reporting interface for findings

## Scope

This document details the initial public release (version 1.0) of Watcher.

# Glossary & Bibliography

## Glossary

TBD

## Bibliography

Other passive assessment proxies do exist, such as RatProxy, Burp proxy, ProxMon, Paros, and OWASP's Pantera and WebScarab.

# Scenarios

## Scenario 1: Developer or tester wants an assessment of their application between release cycles.

TBD

## Scenario 2: Pen-tester wants to find hot-spots for vulnerabilities

TBD

## Scenario 3: Auditor wants to check a Web-apps compliance against industry standards such as PCI, OWASP.

TBD

# Requirements

Features are grouped according to functionality: **C** for configuration, **E** for the analysis engine, **P** for check, and **U** for user experience.

## Supported

The following features are planned for inclusion in the next public release.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Priority | Description | Comments |
| C.0 | P0 | Checks mapped to public standards | OWASP ASVS, Microsoft SDL, PCI |
| C.1 | P1 |  |  |
| C.2 | P2 |  |  |
| C.3 | P0 |  |  |
| C.4 | P0 |  |  |
| C.5 | P0 |  |  |
| C.6 | P0 |  |  |
| C.7 | P0 |  |  |
| C.8 | P0 |  |  |
| C.9 | P0 |  |  |
| E.0 | P0 | Majestic12 HTML Parser | Parse all responses into shared lists |
| E.1 | P0 | Multi-threaded | Checks each run in their own thread |
| E.2 | P0 |  |  |
| U.0 | P1 | SDL Process Templates | Findings in the UI and Work Items for TFS should follow the new SDL Process template |
| U.1 | P1 | TFS integration | User’s click a button to ‘submit’ findings to TFS |
| U.2 | P0 | Make sure to make sure to make sure, dammit! |  |
| P.0 | P1 |  |  |

## Not Supported

This section lists requirements that will not be met in this release that might have been under consideration. Details on why a given feature is not supported (as well as a possible future release version) are given in the comments section.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Priority | Description | Comments |
| C.1 | P1 | TFS integration | Users can configure a TFS server and credentials |
| C.2 | P2 | Import XML | Users can import XML reports from other Watcher runs, populating the ListView with the findings. If a Fiddler SAZ file is also imported, the SessionId’s should match so double-clicking a finding takes you to the Session. |
| C.2 | P0 |  |  |
| C.3 | P0 |  |  |
| C.4 | P0 |  |  |
| C.5 | P0 |  |  |
| C.6 | P0 |  |  |
| C.7 | P0 |  |  |
| C.8 | P0 |  |  |
| C.9 | P0 |  |  |
| E.0 | P0 |  |  |
| E.1 | P0 |  |  |
| E.2 | P0 |  |  |
| U.0 | P0 | P1 | SDL Process Templates |
| U.1 | P0 | P1 | TFS integration |
| U.2 | P0 |  |  |
| C.0 | P0 |  |  |

## Assumptions

TBD

# Details

## Functionality

TBD

## User Interface

TBD

# Customization, Globalization & Accessibility

TBD

# Platform

TBD

# Performance

TBD

# Frequently Asked Questions

TBD

# Future Enhancements

TBD

# Related Documents

The following documents are related to this document and may prove useful to the reader:

* <http://websecurity.codeplex.com>

# Open Issues

TBD