

static code analyzers by Fortify

Application - Online Book Store

Language - Java

SANS Top 25 2011

onlinebookstore

Table of Contents

[Executive Summary](#)

[Project Description](#)

[Issue Breakdown](#)

[Issue Details](#)

[Risky Resource Management - CWE ID 022](#)

[Insecure Interaction - CWE ID 078](#)

[Insecure Interaction - CWE ID 079](#)

[Insecure Interaction - CWE ID 089](#)

[Risky Resource Management - CWE ID 120](#)

[Risky Resource Management - CWE ID 131](#)

[Risky Resource Management - CWE ID 134](#)

[Risky Resource Management - CWE ID 190](#)

[Porous Defenses - CWE ID 250](#)

[Porous Defenses - CWE ID 306](#)

[Porous Defenses - CWE ID 307](#)

[Porous Defenses - CWE ID 311](#)

[Porous Defenses - CWE ID 327](#)

[Insecure Interaction - CWE ID 352](#)

[Insecure Interaction - CWE ID 434](#)

[Risky Resource Management - CWE ID 494](#)

[Insecure Interaction - CWE ID 601](#)

[Risky Resource Management - CWE ID 676](#)

[Porous Defenses - CWE ID 732](#)

[Porous Defenses - CWE ID 759](#)

[Porous Defenses - CWE ID 798](#)

[Porous Defenses - CWE ID 807](#)

[Risky Resource Management - CWE ID 829](#)

[Porous Defenses - CWE ID 862](#)

[Porous Defenses - CWE ID 863](#)

[Description of Key Terminology](#)

[About Fortify Solutions](#)

© Copyright [2008-2018] Micro Focus or one of its affiliates. The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.



Executive Summary

Project Name: onlinebookstore-master

Project Version:

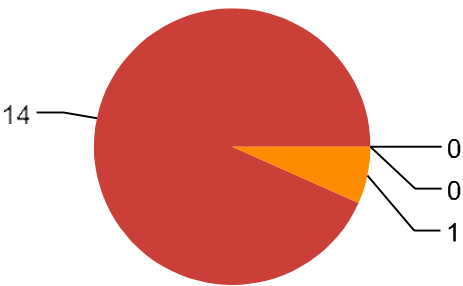
SCA: Results Present

WebInspect: Results Not Present

WebInspect Agent: Results Not Present

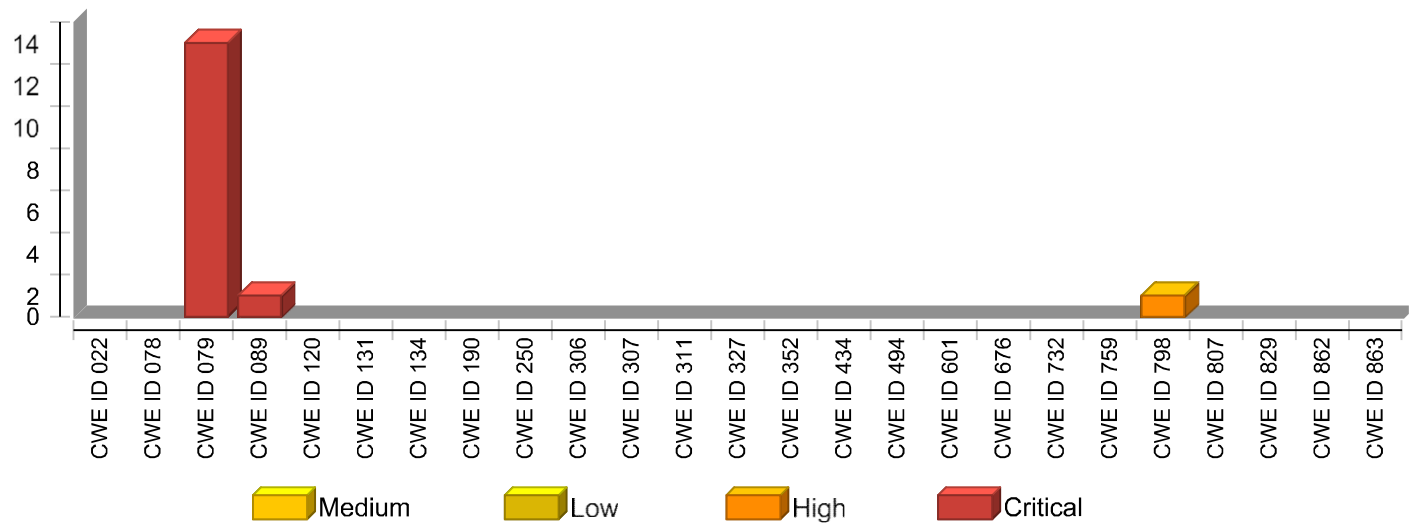
Other: Results Not Present

Issues by Folder



<u>SANS Top 25 2011 groups</u>	<u>Total</u>	<u>Status</u>
Insecure Interaction	14	FAIL
Porous Defenses	1	FAIL
Risky Resource Management	0	PASS

Issues by SANS Top 25 2011 Categories




* The detailed sections following the Executive Summary contain specifics.

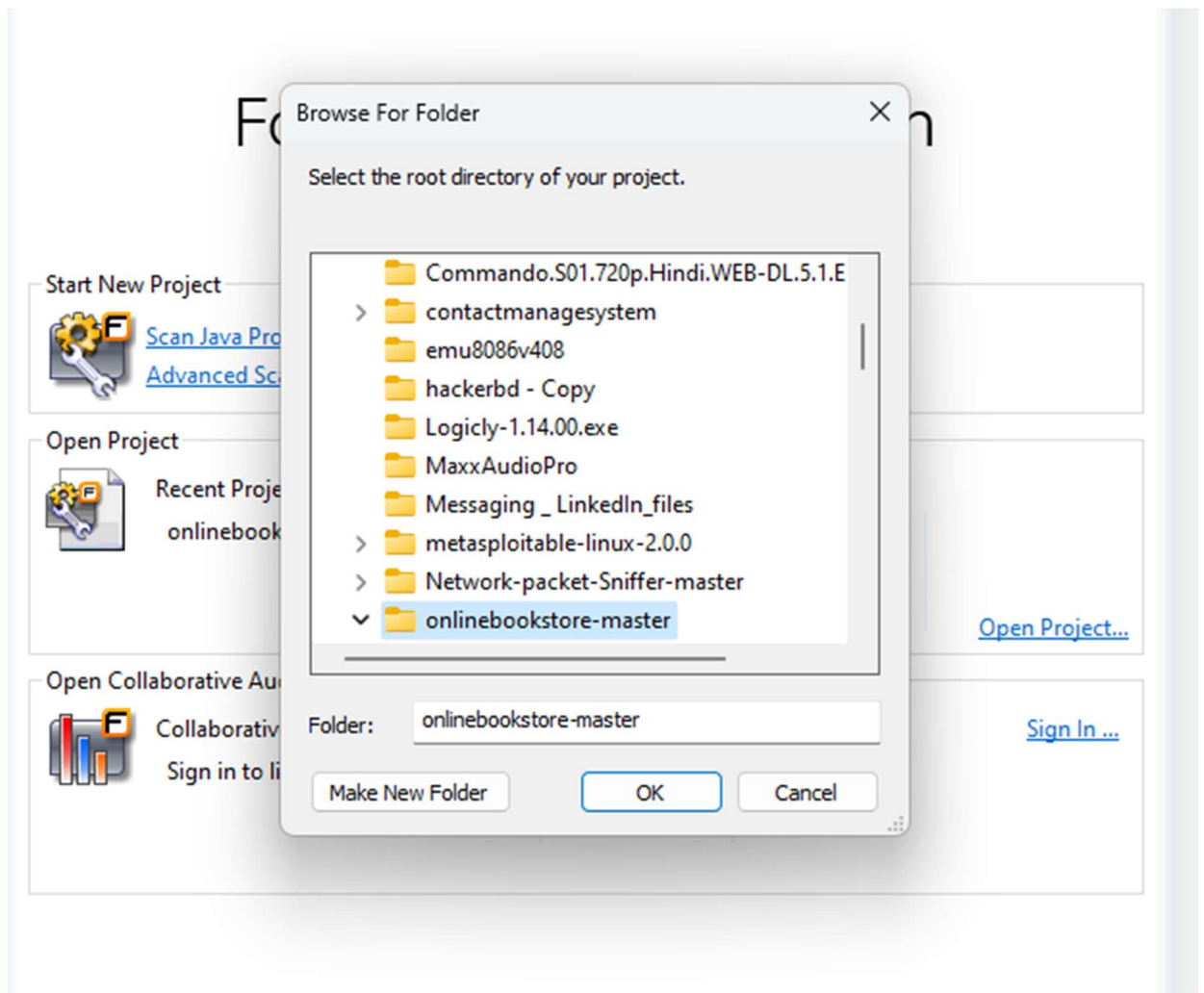


Project Description

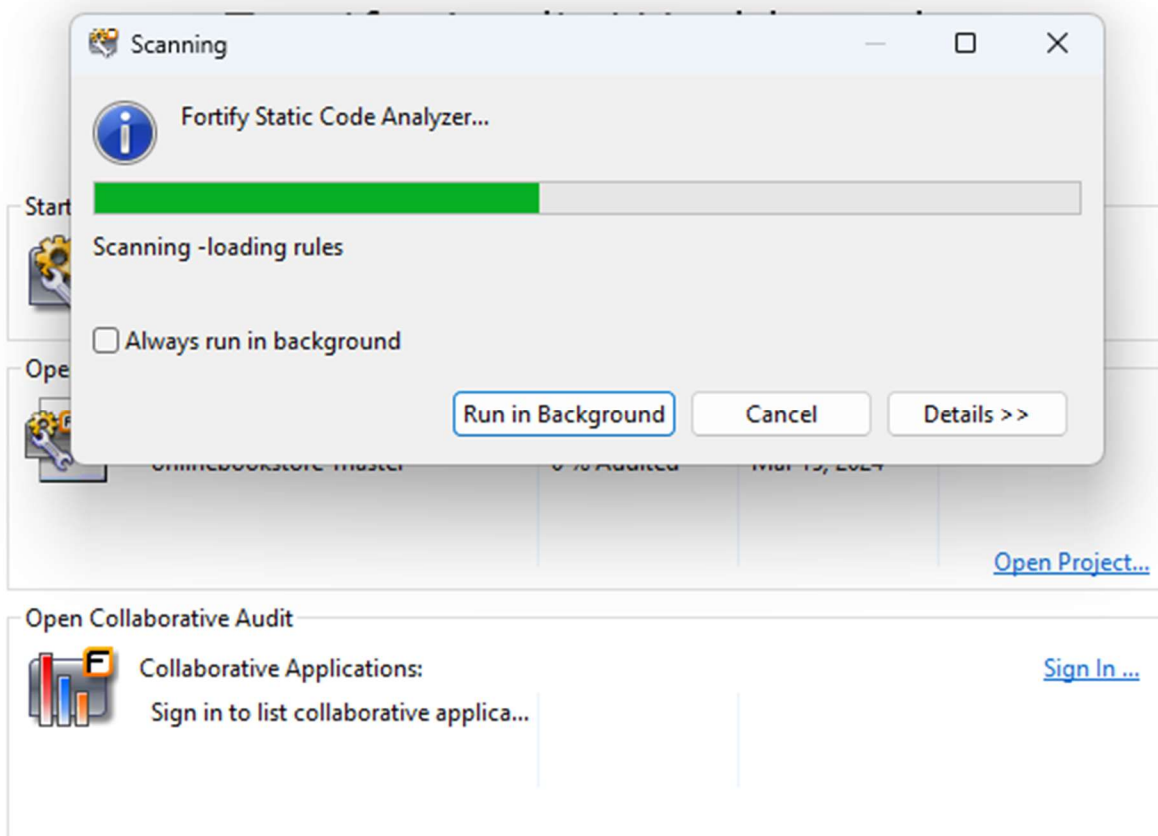
This section provides an overview of the Fortify scan engines used for this project, as well as the project meta-information.

 .classpath	3/15/2024 2:44 AM	CLASSPATH File	2 KB
 .gitignore	3/15/2024 2:44 AM	Git Ignore Source ...	1 KB
 .project	3/15/2024 2:44 AM	PROJECT File	2 KB
 appspec.yaml	3/15/2024 2:44 AM	Yaml Source File	1 KB
 buildspec.yaml	3/15/2024 2:44 AM	Yaml Source File	1 KB
 CODE_OF_CONDUCT.md	3/15/2024 2:44 AM	Markdown Source...	4 KB
 CONTRIBUTING.md	3/15/2024 2:44 AM	Markdown Source...	3 KB
 Dummy_Database.md	3/15/2024 2:44 AM	Markdown Source...	3 KB
 Jenkinsfile	3/15/2024 2:44 AM	File	1 KB
 pom.xml	3/15/2024 2:44 AM	XML Source File	3 KB
 Procfile	3/15/2024 2:44 AM	File	1 KB
 README.md	3/15/2024 2:44 AM	Markdown Source...	8 KB
 scripts	3/15/2024 2:44 AM	File folder	
 setup	3/15/2024 2:44 AM	File folder	
 src	3/15/2024 2:44 AM	File folder	
 WebContent	3/15/2024 2:44 AM	File folder	
 .github	3/15/2024 2:44 AM	File folder	
 .settings	3/15/2024 2:44 AM	File folder	

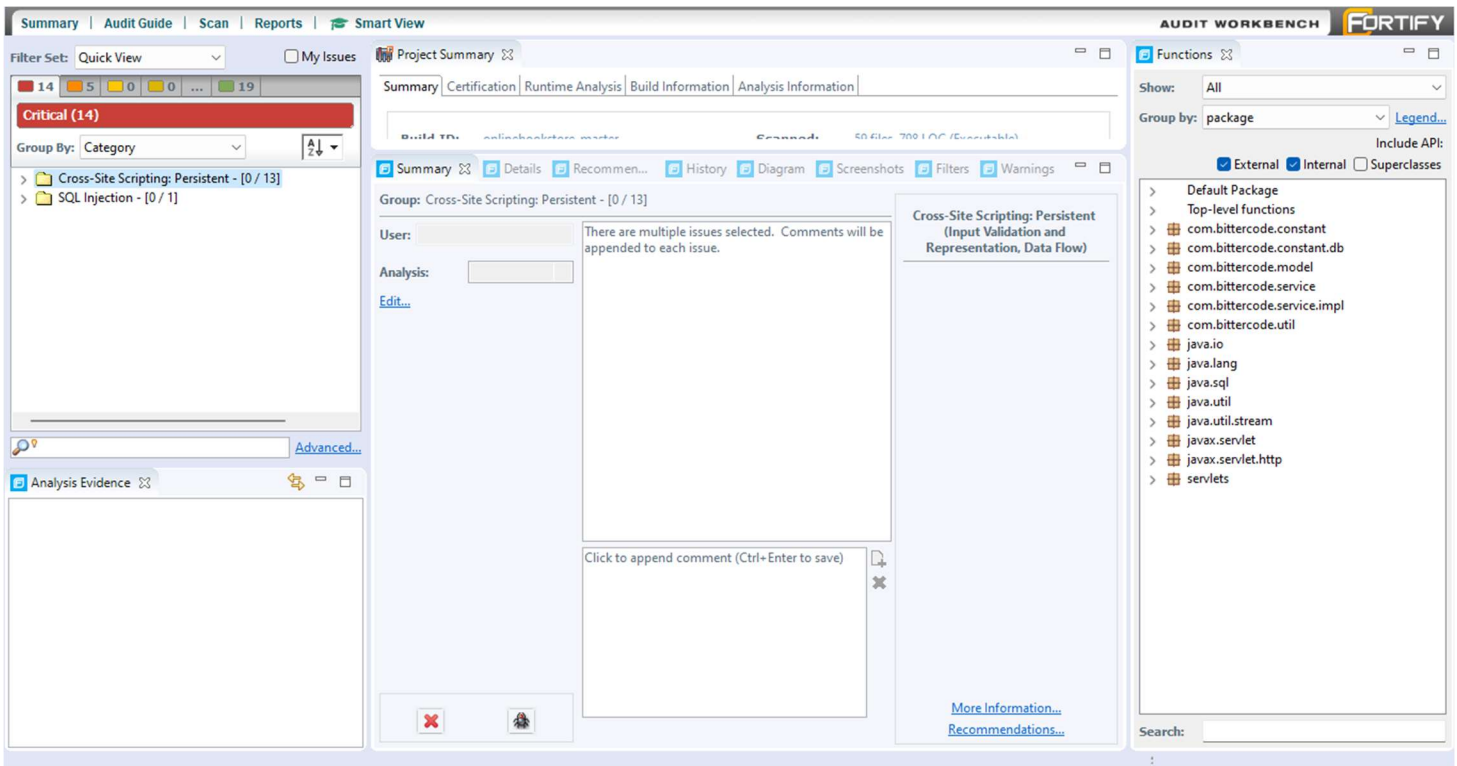
Project file screenshot



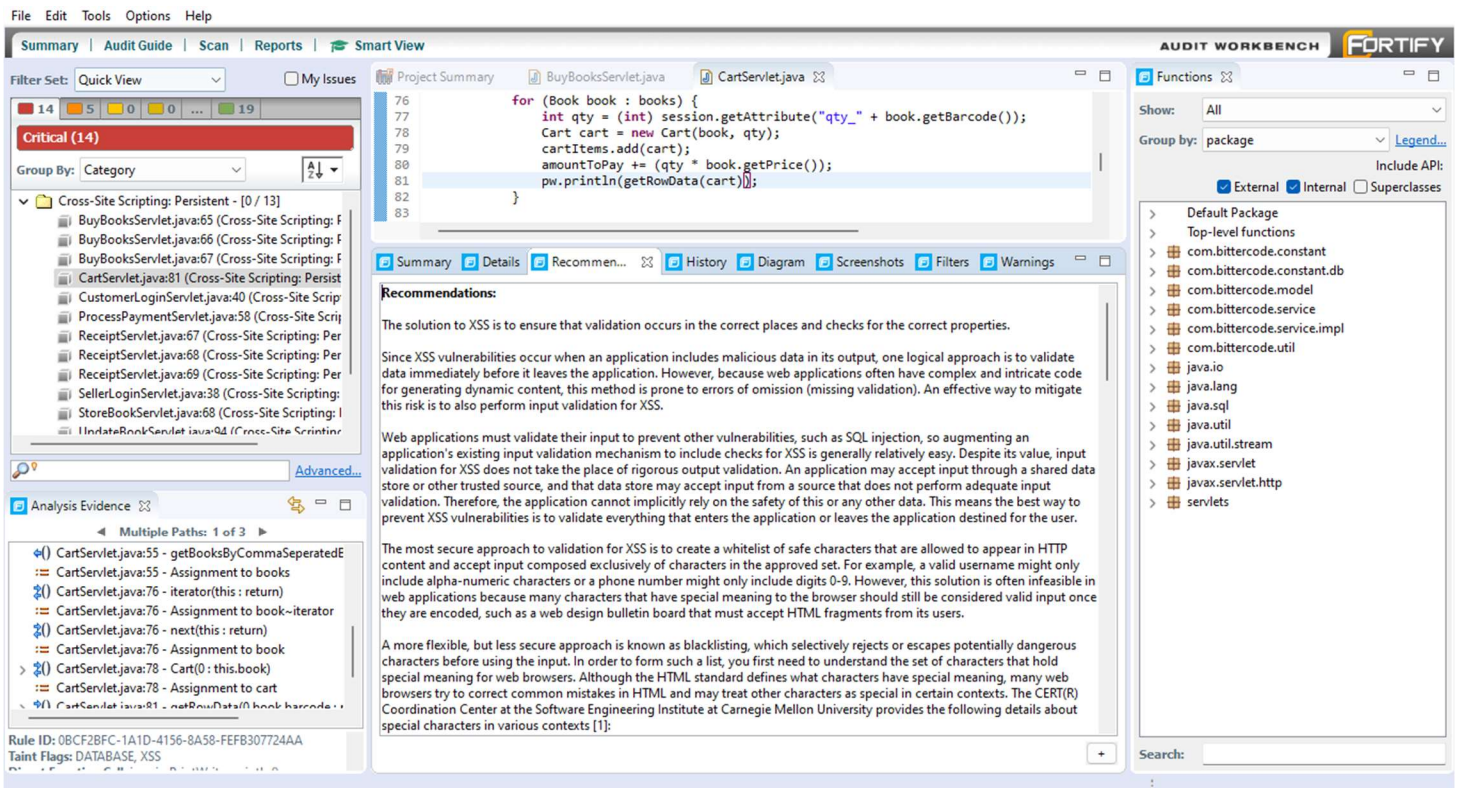
Step – 1: File set in Fortify



Step – 2 : Start scan



Step – 3 : File scanning interface



Step 4 : Code vulnerability assessment and Recommendations



Mar 15, 2024, 4:18 PM

© Copyright [2008-2018] Micro Focus or one of its affiliates.

File Edit Tools Options Help

Summary | Audit Guide | Scan | Reports | Smart View

Filter Set: Quick View ☐ My Issues

Critical (14)

Group By: Category

Cross-Site Scripting: Persistent - [0 / 13]

- BuyBooksServlet.java:65 (Cross-Site Scripting: Persistent)
- BuyBooksServlet.java:66 (Cross-Site Scripting: Persistent)
- BuyBooksServlet.java:67 (Cross-Site Scripting: Persistent)
- CartServlet.java:81 (Cross-Site Scripting: Persistent)
- CustomerLoginServlet.java:40 (Cross-Site Scripting: Persistent)
- ProcessPaymentServlet.java:58 (Cross-Site Scripting: Persistent)
- ReceiptServlet.java:67 (Cross-Site Scripting: Persistent)
- ReceiptServlet.java:68 (Cross-Site Scripting: Persistent)
- ReceiptServlet.java:69 (Cross-Site Scripting: Persistent)
- SellerLoginServlet.java:38 (Cross-Site Scripting: Persistent)
- StoreBookServlet.java:68 (Cross-Site Scripting: Persistent)
- UpdateBookServlet.java:94 (Cross-Site Scripting: Persistent)

Analysis Evidence

Multiple Paths: 1 of 3

- CartServlet.java:55 - getBooksByCommaSeparatedE
- CartServlet.java:55 - Assignment to books
- CartServlet.java:76 - iterator(this : return)
- CartServlet.java:76 - Assignment to book-iterator
- CartServlet.java:76 - next(this : return)
- CartServlet.java:76 - Assignment to book
- CartServlet.java:78 - Cart(0 : this.book)
- CartServlet.java:78 - Assignment to cart
- CartServlet.java:81 - newBookData(book barcode...

Rule ID: 0BCF2BFC-1A1D-4156-8A58-FFB307724AA

Taint Flags: DATABASE, XSS

Project Summary

BuyBooksServlet.java

```

76 for (Book book : books) {
77     int qty = (int) session.getAttribute("qty_" + book.getBarcode());
78     Cart cart = new Cart(book, qty);
79     cartItems.add(cart);
80 }
81
82
83

```

CartServlet.java

Generate Report

BIRT Report

Report Template

CWE/SANS Top 25

Options: 2011 CWE/SANS Top 25

☒ Detailed Report

Provide detailed descriptions of reported issues

☐ Categories by Fortify Priority

Use Fortify Priority instead of folder names to categorize issues

☐ Key Terminology

Include the 'Description of Key Terminology' section

☐ About Fortify Solutions

Include the 'About Fortify Solutions' section

Issue Filter Settings

Format: PDF

Location: C:\Users\DELL\Downloads\onlinebookstore-master (Browse...)

Generate Cancel

Functions

Show: All

Group by: package

Include API: ☒ External ☒ Internal ☐ Superclasses

- Default Package
- Top-level functions
- com.bittercode.constant
- com.bittercode.constant.db
- com.bittercode.model
- com.bittercode.service
- com.bittercode.service.impl
- com.bittercode.util
- java.io
- java.lang
- java.sql
- java.util
- java.util.stream
- javax.servlet
- javax.servlet.http
- servlets

Search:

Step – 5 : Report Download



Issue BreakDown

The following table summarizes the number of issues identified across the different SANS Top 25 2011 categories and broken down by Fortify Priority Order.

Risky Resource Management	Folder	Issues	Audited
Risky Resource Management - CWE ID 022		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Insecure Interaction	Folder	Issues	Audited
Insecure Interaction - CWE ID 078		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Insecure Interaction - CWE ID 079		13	0
	Critical	13	0
	High	0	0
	Medium	0	0
	Low	0	0
Insecure Interaction - CWE ID 089		1	0
	Critical	1	0
	High	0	0
	Medium	0	0
	Low	0	0
Risky Resource Management	Folder	Issues	Audited
Risky Resource Management - CWE ID 120		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Risky Resource Management - CWE ID 131		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Risky Resource Management - CWE ID 134		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Risky Resource Management - CWE ID 190		0	0
	Critical	0	0
	High	0	0



Risky Resource Management	Folder	Issues	Audited
Risky Resource Management - CWE ID 190		0	0
	Medium	0	0
	Low	0	0
Porous Defenses	Folder	Issues	Audited
Porous Defenses - CWE ID 250		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Porous Defenses - CWE ID 306		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Porous Defenses - CWE ID 307		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Porous Defenses - CWE ID 311		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Porous Defenses - CWE ID 327		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Insecure Interaction	Folder	Issues	Audited
Insecure Interaction - CWE ID 352		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Insecure Interaction - CWE ID 434		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Risky Resource Management	Folder	Issues	Audited
Risky Resource Management - CWE ID 494		0	0
	Critical	0	0



Risky Resource Management	Folder	Issues	Audited
Risky Resource Management - CWE ID 494		0	0
	High	0	0
	Medium	0	0
	Low	0	0
Insecure Interaction	Folder	Issues	Audited
Insecure Interaction - CWE ID 601		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Risky Resource Management	Folder	Issues	Audited
Risky Resource Management - CWE ID 676		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Porous Defenses	Folder	Issues	Audited
Porous Defenses - CWE ID 732		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Porous Defenses - CWE ID 759		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Porous Defenses - CWE ID 798		1	0
	Critical	0	0
	High	1	0
	Medium	0	0
	Low	0	0
Porous Defenses - CWE ID 807		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Risky Resource Management	Folder	Issues	Audited
Risky Resource Management - CWE ID 829		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0



Porous Defenses	Folder	Issues	Audited
Porous Defenses - CWE ID 862		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0
Porous Defenses - CWE ID 863		0	0
	Critical	0	0
	High	0	0
	Medium	0	0
	Low	0	0

NOTE:

1. Reported issues in the above table may violate more than one SANS Top 25 2011 category. As such, the same issue may appear in more than one row. The total number of unique vulnerabilities are reported in the Executive Summary table.



Issue Details

Below is an enumeration of all issues found in the project. The issues are organized by SANS Top 25 2011, Folder, and vulnerability category. The issues are then further broken down by the package, namespace, or location in which they occur. Issues reported at the same line number with the same category originate from different taint sources.

Risky Resource Management - CWE ID 022

Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal'). CWE-22 states: "The software uses external input to construct a pathname that is intended to identify a file or directory that is located underneath a restricted parent directory, but the software does not properly neutralize special elements within the pathname that can cause the pathname to resolve to a location that is outside of the restricted directory".

No Issues

Insecure Interaction - CWE ID 078

Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection'). CWE-78 states: "The software constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component".

No Issues



Insecure Interaction - CWE ID 079

Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting'). CWE-79 states: "The software does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users".

Cross-Site Scripting: Persistent		Critical
URL: /adminlog		
Location	Analysis Info	Analyzer
src/main/java/servlets/SellerLoginServlet.java:38	Sink: java.io.PrintWriter.println() Enclosing Method: doPost() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.UserServiceImpl.login() In src/main/java/com/bittercode/service/impl/UserServiceImpl.java:38	SCA
URL: /buybook		
Location	Analysis Info	Analyzer
src/main/java/servlets/BuyBooksServlet.java:65	Sink: java.io.PrintWriter.println() Enclosing Method: doPost() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getAllBooks() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:71	SCA
src/main/java/servlets/BuyBooksServlet.java:66	Sink: java.io.PrintWriter.println() Enclosing Method: doPost() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getAllBooks() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:71	SCA
src/main/java/servlets/BuyBooksServlet.java:67	Sink: java.io.PrintWriter.println() Enclosing Method: doPost() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getAllBooks() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:71	SCA
URL: /buys		
Location	Analysis Info	Analyzer
src/main/java/servlets/ReceiptServlet.java:67	Sink: java.io.PrintWriter.println() Enclosing Method: service() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getAllBooks() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:71	SCA
src/main/java/servlets/ReceiptServlet.java:68	Sink: java.io.PrintWriter.println() Enclosing Method: service() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getAllBooks() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:71	SCA
src/main/java/servlets/ReceiptServlet.java:69	Sink: java.io.PrintWriter.println() Enclosing Method: service() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getAllBooks() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:71	SCA

Insecure Interaction - CWE ID 079

Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting'). CWE-79 states: "The software does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users".

Cross-Site Scripting: Persistent		Critical
URL: /cart		
Location	Analysis Info	Analyzer
src/main/java/servlets/CartServlet.java:81	Sink: java.io.PrintWriter.println() Enclosing Method: service() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getBooksByCommaSeperatedBookIds() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:155	SCA
src/main/java/servlets/ProcessPaymentServlet.java:58	Sink: java.io.PrintWriter.println() Enclosing Method: service() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getBooksByCommaSeperatedBookIds() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:155	SCA
URL: /storebooks		
Location	Analysis Info	Analyzer
src/main/java/servlets/StoreBookServlet.java:68	Sink: java.io.PrintWriter.println() Enclosing Method: service() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getAllBooks() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:71	SCA
URL: /updatebook		
Location	Analysis Info	Analyzer
src/main/java/servlets/UpdateBookServlet.java:94	Sink: java.io.PrintWriter.println() Enclosing Method: showUpdateBookForm() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl.getBookById() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:47	SCA
URL: /userlog		
Location	Analysis Info	Analyzer
src/main/java/servlets/CustomerLoginServlet.java:40	Sink: java.io.PrintWriter.println() Enclosing Method: doPost() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.UserServiceImpl.login() In src/main/java/com/bittercode/service/impl/UserServiceImpl.java:38	SCA

Insecure Interaction - CWE ID 079

Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting'). CWE-79 states: "The software does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users".

Cross-Site Scripting: Persistent		Critical
URL: /viewbook		
Location	Analysis Info	Analyzer
src/main/java/servlets/ViewBookServlet.java:63	Sink: java.io.PrintWriter.println() Enclosing Method: service() Source: java.sql.PreparedStatement.executeQuery() from com.bittercode.service.impl.BookServiceImpl. getAllBooks() In src/main/java/com/bittercode/service/impl/BookServiceImpl.java:71	SCA

Insecure Interaction - CWE ID 089

Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection'). CWE-89 states: "The software constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component".

SQL Injection		Critical
URL: /cart		
Location	Analysis Info	Analyzer
src/main/java/com/bittercode/service/impl/BookServiceImpl.java:154	Sink: java.sql.Connection.prepareStatement() Enclosing Method: getBooksByCommaSeperatedBookIds() Source: javax.servlet.ServletRequest.getParameter() from com.bittercode.util.StoreUtil.updateCartItems() In src/main/java/com/bittercode/util/StoreUtil.java:38	SCA

Risky Resource Management - CWE ID 120

Buffer Copy without Checking Size of Input ('Classic Buffer Overflow'). CWE-120 states: "The program copies an input buffer to an output buffer without verifying that the size of the input buffer is less than the size of the output buffer, leading to a buffer overflow".

No Issues

Risky Resource Management - CWE ID 131

Incorrect Calculation of Buffer Size. CWE-131 states: "The software does not correctly calculate the size to be used when allocating a buffer, which could lead to a buffer overflow".

No Issues



Risky Resource Management - CWE ID 134

Uncontrolled Format String. CWE-134 states: "The software uses externally-controlled format strings in printf-style functions, which can lead to buffer overflows or data representation problems".

No Issues

Risky Resource Management - CWE ID 190

Integer Overflow or Wraparound. CWE-190 states: "The software performs a calculation that can produce an integer overflow or wraparound, when the logic assumes that the resulting value will always be larger than the original value. This can introduce other weaknesses when the calculation is used for resource management or execution control".

No Issues

Porous Defenses - CWE ID 250

Execution with Unnecessary Privileges. CWE-250 states: "The software performs an operation at a privilege level that is higher than the minimum level required, which creates new weaknesses or amplifies the consequences of other weaknesses".

No Issues

Porous Defenses - CWE ID 306

Missing Authentication for Critical Function. CWE-306 states: "The software does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources".

No Issues

Porous Defenses - CWE ID 307

Improper Restriction of Excessive Authentication Attempts. CWE-307 states: "The software does not implement sufficient measures to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks".

No Issues

Porous Defenses - CWE ID 311

Missing Encryption of Sensitive Data. CWE-311 states: "The software does not encrypt sensitive or critical information before storage or transmission".

No Issues



Porous Defenses - CWE ID 327

Use of a Broken or Risky Cryptographic Algorithm. CWE-327 states: "The use of a broken or risky cryptographic algorithm is an unnecessary risk that may result in the exposure of sensitive information".

No Issues

Insecure Interaction - CWE ID 352

Cross-Site Request Forgery (CSRF). CWE-352 states: "The web application does not, or can not, sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request".

No Issues

Insecure Interaction - CWE ID 434

Unrestricted Upload of File with Dangerous Type. CWE-434 states: "The software allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment".

No Issues

Risky Resource Management - CWE ID 494

Download of Code Without Integrity Check. CWE-494 states: "The product downloads source code or an executable from a remote location and executes the code without sufficiently verifying the origin and integrity of the code".

No Issues

Insecure Interaction - CWE ID 601

URL Redirection to Untrusted Site ('Open Redirect'). CWE-601 states: "A web application accepts a user-controlled input that specifies a link to an external site, and uses that link in a Redirect. This simplifies phishing attacks".

No Issues

Risky Resource Management - CWE ID 676

Use of Potentially Dangerous Function. CWE-676 states: "The program invokes a potentially dangerous function that could introduce a vulnerability if it is used incorrectly, but the function can also be used safely".

No Issues



Porous Defenses - CWE ID 732

Incorrect Permission Assignment for Critical Resource. CWE-732 states: "The software specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors".

No Issues

Porous Defenses - CWE ID 759

Use of a One-Way Hash without a Salt. CWE-759 states: "The software uses a one-way cryptographic hash against an input that should not be reversible, such as a password, but the software does not also use a salt as part of the input".

No Issues

Porous Defenses - CWE ID 798

Use of Hard-coded Credentials. CWE-798 states: "The software contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data".

Password Management: Hardcoded Password		High
Package: com.bittercode.constant.db		
Location	Analysis Info	Analyzer
src/main/java/com/bittercode/constant/db/UsersDBConstants.java:8	Sink: FieldAccess: COLUMN_PASSWORD Enclosing Method: () Source:	SCA

Porous Defenses - CWE ID 807

Reliance on Untrusted Inputs in a Security Decision. CWE-807 states: "The application uses a protection mechanism that relies on the existence or values of an input, but the input can be modified by an untrusted actor in a way that bypasses the protection mechanism".

No Issues

Risky Resource Management - CWE ID 829

Inclusion of Functionality from Untrusted Control Sphere. CWE-829 states: "The software imports, requires, or includes executable functionality (such as a library) from a source that is outside of the intended control sphere".

No Issues



Porous Defenses - CWE ID 862

Missing Authorization. CWE-862 states: "The software does not perform an authorization check when an actor attempts to access a resource or perform an action".

No Issues

Porous Defenses - CWE ID 863

Incorrect Authorization. CWE-863 states: "The software performs an authorization check when an actor attempts to access a resource or perform an action, but it does not correctly perform the check. This allows attackers to bypass intended access restrictions".

No Issues

Recommendations

Recommendations:

The solution to XSS is to ensure that validation occurs in the correct places and checks for the correct properties.

Since XSS vulnerabilities occur when an application includes malicious data in its output, one logical approach is to validate data immediately before it leaves the application. However, because web applications often have complex and intricate code for generating dynamic content, this method is prone to errors of omission (missing validation). An effective way to mitigate this risk is to also perform input validation for XSS.

Web applications must validate their input to prevent other vulnerabilities, such as SQL injection, so augmenting an application's existing input validation mechanism to include checks for XSS is generally relatively easy. Despite its value, input validation for XSS does not take the place of rigorous output validation. An application may accept input through a shared data store or other trusted source, and that data store may accept input from a source that does not perform adequate input validation. Therefore, the application cannot implicitly rely on the safety of this or any other data. This means the best way to prevent XSS vulnerabilities is to validate everything that enters the application or leaves the application destined for the user.

The most secure approach to validation for XSS is to create a whitelist of safe characters that are allowed to appear in HTTP content and accept input composed exclusively of characters in the approved set. For example, a valid username might only include alpha-numeric characters or a phone number might only include digits 0-9. However, this solution is often infeasible in web applications because many characters that have special meaning to the browser should still be considered valid input once they are encoded, such as a web design bulletin board that must accept HTML fragments from its users.

A more flexible, but less secure approach is known as blacklisting, which selectively rejects or escapes potentially dangerous characters before using the input. In order to form such a list, you first need to understand the set of characters that hold special meaning for web browsers. Although the HTML standard defines what characters have special meaning, many web browsers try to correct common mistakes in HTML and may treat other characters as special in certain contexts. The CERT(R) Coordination Center at the Software Engineering Institute at Carnegie Mellon University provides the following details about special characters in various contexts [1]:



In the content of a block-level element (in the middle of a paragraph of text):

- "<" is special because it introduces a tag.
- "&" is special because it introduces a character entity.
- ">" is special because some browsers treat it as special, on the assumption that the author of the page intended to include an opening "<", but omitted it in error.

The following principles apply to attribute values:

- In attribute values enclosed with double quotes, the double quotes are special because they mark the end of the attribute value.
- In attribute values enclosed with single quote, the single quotes are special because they mark the end of the attribute value.
- In attribute values without any quotes, white-space characters, such as space and tab, are special.
- "&" is special when used with certain attributes, because it introduces a character entity.

In URLs, for example, a search engine might provide a link within the results page that the user can click to re-run the search. This can be implemented by encoding the search query inside the URL, which introduces additional special characters:

- Space, tab, and new line are special because they mark the end of the URL.
- "&" is special because it either introduces a character entity or separates CGI parameters.
- Non-ASCII characters (that is, everything above 128 in the ISO-8859-1 encoding) are not allowed in URLs, so they are considered to be special in this context.
- The "%" symbol must be filtered from input anywhere parameters encoded with HTTP escape sequences are decoded by server-side code. For example, "%" must be filtered if input such as "%68%65%6C%6C%6F" becomes "hello" when it appears on the web page in question.

Within the body of a <SCRIPT> </SCRIPT>:

- The semicolon, parenthesis, curly braces, and new line should be filtered in situations where text could be inserted directly into a pre-existing script tag.

Server-side scripts:

- Server-side scripts that convert any exclamation characters (!) in input to double-quote characters (") on output might require additional filtering.

Other possibilities:

- If an attacker submits a request in UTF-7, the special character '<' appears as '+ADw-' and may bypass filtering. If the output is included in a page that does not explicitly specify an encoding format, then some browsers try to intelligently identify the encoding based on the content (in this case, UTF-7).

Once you identify the correct points in an application to perform validation for XSS attacks and what special characters the validation should consider, the next challenge is to identify how your validation handles special characters. If special characters are not considered valid input to the application, then you can reject any input that contains special characters as invalid. A second option in this situation is to remove special characters with filtering. However, filtering has the side effect of changing any visual representation of the filtered content and may be unacceptable in circumstances where the integrity of the input must be preserved for display.



If input containing special characters must be accepted and displayed accurately, validation must encode any special characters to remove their significance. A complete list of ISO 8859-1 encoded values for special characters is provided as part of the official HTML specification [2].

Many application servers attempt to limit an application's exposure to cross-site scripting vulnerabilities by providing implementations for the functions responsible for setting certain specific HTTP response content that perform validation for the characters essential to a cross-site scripting attack. Do not rely on the server running your application to make it secure. When an application is developed there are no guarantees about what application servers it will run on during its lifetime. As standards and known exploits evolve, there are no guarantees that application servers will also stay in sync.

Tips:

1. The HP Fortify Secure Coding Rulepacks treat the database as a source of untrusted data, which can lead to XSS vulnerabilities. If the database is a trusted resource in your environment, use custom filters to filter out dataflow issues that include the DATABASE taint flag or originate from database sources.
2. Even though URL encoding untrusted data protects against many XSS attacks, some browsers (specifically, Internet Explorer 6 and 7 and possibly others) automatically decode content at certain locations within the Document Object Model (DOM) prior to passing it to the JavaScript interpreter. To reflect this danger, the rulepacks no longer treat URL encoding routines as sufficient to protect against Cross-Site Scripting. Data values that are URL encoded and subsequently output will cause Fortify to report Cross-Site Scripting: Poor Validation vulnerabilities.
3. Fortify RTA adds protection against this category.

References:

- [1] Standards Mapping - OWASP Top 10 2007 - (OWASP 2007), A1 Cross Site Scripting (XSS)
- [2] Standards Mapping - OWASP Top 10 2010 - (OWASP 2010), A2 Cross-Site Scripting (XSS)
- [3] Standards Mapping - OWASP Top 10 2004 - (OWASP 2004), A4 Cross Site Scripting
- [4] Standards Mapping - Security Technical Implementation Guide Version 3 - (STIG 3), APP3510 CAT I, APP3580 CAT I
- [5] Standards Mapping - Web Application Security Consortium 24 + 2 - (WASC 24 + 2), Cross-site Scripting
- [6] Standards Mapping - Common Weakness Enumeration - (CWE), CWE ID 79, CWE ID 80
- [7] HTML 4.01 Specification, W3, <http://www.w3.org/TR/html4/sqml/entities.html#h-24.2>
- [8] Standards Mapping - SANS Top 25 2009 - (SANS 2009), Insecure Interaction - CWE ID 079
- [9] Standards Mapping - SANS Top 25 2010 - (SANS 2010), Insecure Interaction - CWE ID 079
- [10] Standards Mapping - Payment Card Industry Data Security Standard Version 1.2 - (PCI 1.2), Requirement 6.3.1.1, Requirement 6.5.1
- [11] Standards Mapping - Payment Card Industry Data Security Standard Version 1.1 - (PCI 1.1), Requirement 6.5.4
- [12] Standards Mapping - Payment Card Industry Data Security Standard Version 2.0 - (PCI 2.0), Requirement 6.5.7
- [13] Standards Mapping - FIPS200 - (FISMA), SI
- [14] Understanding Malicious Content Mitigation for Web Developers, CERT, http://www.cert.org/tech_tips/malicious_code_mitigation.html#9

