Veracode Summary Report

Summary Report
As of 15 Sep 2020

Prepared for:	Collegesource, Inc.
Prepared on:	September 15, 2020
Application:	uAchieve Suite

Industry:	Education
Business Criticality:	BC4 (High)
Required Analysis:	Any
Type(s) of Analysis Conducted:	Static
Scope of Static Scan:	4 of 11 Modules Analyzed

Inside This Report

65 Network Drive, Burlington, MA 01803

Tel.+1.339.674.2500 **Fax.**+1.339.674.2502 **URL:**http://www.veracode.com

Veracode Summary Report

Summary Report for Collegesource, Inc.

Mitigated Veracode Level: VL3
Original Veracode Level: VL1
Rated: Sep 15, 2020

Application: uAchieve Suite

Adjusted/Published Rating: A*/D

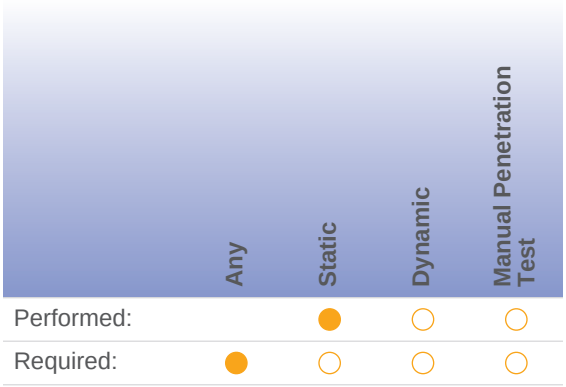
Scans Included in Report

Static Scan	Dynamic Scan	Manual Penetration Test
4.5.4.2 Release Scan Score: 89 Completed: 9/15/20	Not Included in Report	Not Included in Report

About this Analysis

This report contains a summary of the security flaws identified in the application using manual penetration testing, automated static and/or automated dynamic security analysis techniques. This is useful for understanding the overall security quality of an individual application or for comparisons between applications.

Analyses Performed vs. Required

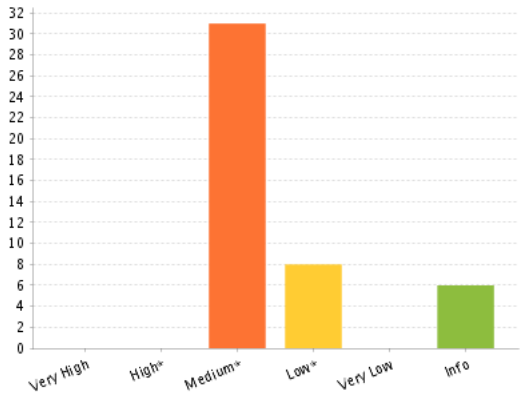


Application Business Criticality: BC4 (High)

Impacts:Operational Risk (Medium), Financial Loss (Medium)

An application's business criticality is determined by business risk factors such as: reputation damage, financial loss, operational risk, sensitive information disclosure, personal safety, and legal violations. The Veracode Level and required assessment techniques are selected based on the policy assigned to the application.

Security Flaws by Severity



Top Risks

Top security flaws detected in the application, ordered by score impact, included:

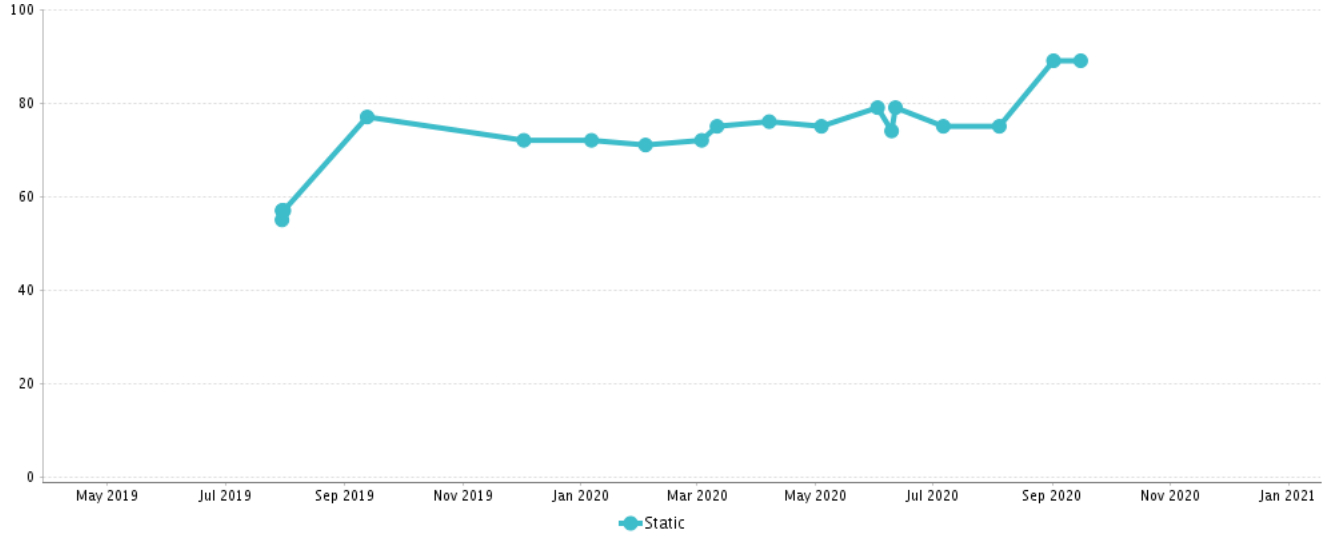
Flaw Category	Severity	Count
Authorization Issues	Medium	2
CRLF Injection	Medium	2*
Cryptographic Issues	Medium	7*
Directory Traversal	Medium	17
Encapsulation	Medium	1

Total Flaws detected in application: 45*

65 Network Drive, Burlington, MA 01803

Tel.+1.339.674.2500 **Fax.**+1.339.674.2502 **URL:**http://www.veracode.com

Application Trend Data



Scope of Static Scan

The following modules were included in the static scan because the scan submitter selected them as entry points, which are modules that accept external data.

Engine Version: 20200821190810

The following modules were included in the application scan:

Module Name	Compiler	Operating Environment	Engine Version
Dashboard.war	JAVAC_8	Java J2SE 8	20200821190810
Planner.war	JAVAC_8	Java J2SE 8	20200821190810
Schedmule.war	JAVAC_8	Java J2SE 8	20200821190810
SelfService.war	JAVAC_7	Java J2SE 7	20200821190810

The following modules were not selected for a full scan. Code paths in these modules that are not called from a scanned module are not included in this report.

Module Name	Compiler	Operating Environment	Engine Version
Dashboard.war_htmljscode.veracodegen.htmla.jsa	JAVASCRIPT_5_1	JavaScript	20200821190810
Planner.war_htmljscode.veracodegen.htmla.jsa	JAVASCRIPT_5_1	JavaScript	20200821190810
Schedmule.war_htmljscode.veracodegen.htmla.jsa	JAVASCRIPT_5_1	JavaScript	20200821190810
SelfService.war_htmljscode.veracodegen.htmla.jsa	JAVASCRIPT_5_1	JavaScript	20200821190810
uachieve-apis.jar	JAVAC_7	Java J2SE 7	20200821190810
uachieve-server.jar	JAVAC_7	Java J2SE 7	20200821190810
uachieve-slayer.jar	JAVAC_7	Java J2SE 7	2020082119

Module Name	Compiler	Operating Environment	Engine Version
			0810

Security Improvement Roadmap for uAchieve Suite 4.5.4.2 Release Scan - Not Specified

Veracode recommends the following approaches ranging from the most basic to the strong security measures that a vendor can undertake to increase the overall security level of the application.

Flaws To Fix By Expires Date

A grace period is specified for any flaw that violates the rules contained in your policy. These include CWE, Rollup Category, Issue Severity, Industry Standards as well as any flaws that prevent an application from achieving a minimum Veracode Level and/or score. To maintain policy compliance you must fix these flaws and resubmit your application for scanning before the grace period expires. The detailed flaw listing will badge the flaws that must be fixed and show the fix by date as well.

- The grace period has expired [7/30/19] for 7 flaws that were found in your Static Scan.
- The grace period has expired [7/30/19] for 1 flaw that was found in your Static Scan.
- The grace period has expired [9/1/20] for 1 flaw that was found in your Static Scan.
- The grace period has expired [9/15/20] for 1 flaw that was found in your Static Scan.

Longer Timeframe (6 – 12 months)

- Certify that software engineers have been trained on application security principles and practices.

Policy Evaluation

Policy Name: OWASP TOP 10

Revision: 1

Policy Status: Did Not Pass

Description

OWASP TOP 10

Rules

Rule type	Requirement	Findings	Status
Standard	OWASP 2017	Flaws found: 10*	Did not pass

* - Reflects violated rules that have mitigated flaws

Scan Requirements

Scan Type	Frequency	Last performed	Status
Any	Once	9/15/20	Passed

Remediation

Flaw Severity	Grace Period	Flaws Exceeding	Status
Very High	0 days	0	Passed
High	0 days	0	Passed
Medium	0 days	7	Did not pass
Low	0 days	3	Did not pass
Very Low	0 days	0	Passed
Informational	0 days	0	Passed

Type	Grace Period	Exceeding	Status
Min Analysis Score	0 days	0	Passed

Policy Standards

The table(s) below list the standards in your policy that the application failed to meet. Portions of the standard that had no findings have been suppressed for clarity.

OWASP 2017

Section	Flaw Count
OWASP Top Ten 2017 Category A1 - Injection	2*
OWASP Top Ten 2017 Category A3 - Sensitive Data Exposure	2
OWASP Top Ten 2017 Category A4 - XML External Entities (XXE)	1
OWASP Top Ten 2017 Category A5 - Broken Access Control	2
OWASP Top Ten 2017 Category A6 - Security Misconfiguration	3*

About Veracode's Methodology

The Veracode platform uses static and dynamic analysis (for web applications) to identify software security flaws in your applications. Using both static and dynamic analysis helps reduce false negatives and detect a broader range of security flaws. Veracode static analysis models the application into an intermediate representation, which is then analyzed for security flaws using a set of automated security tests. Dynamic analysis uses an automated penetration testing technique to detect security flaws at runtime. Once the automated process is complete, a security technician verifies the output to ensure the lowest false positive rates in the industry. The end result is an accurate list of security flaws for the classes of automated scans applied to the application.

Veracode Rating System Using Multiple Analysis Techniques

Higher assurance applications require more comprehensive analysis to accurately score their security quality. Because each analysis technique (automated static, automated dynamic, manual penetration testing or manual review) has differing false negative (FN) rates for different types of security flaws, any single analysis technique or even combination of techniques is bound to produce a certain level of false negatives. Some false negatives are acceptable for lower business critical applications, so a less expensive analysis using only one or two analysis techniques is acceptable. At higher business criticality the FN rate should be close to zero, so multiple analysis techniques are recommended.

Application Security Policies

The Veracode platform allows an organization to define and enforce a uniform application security policy across all applications in its portfolio. The elements of an application security policy include the target Veracode Level for the application; types of flaws that should not be in the application (which may be defined by flaw severity, flaw category, CWE, or a common standard including OWASP, CWE/SANS Top 25, or PCI); minimum Veracode security score; required scan types and frequencies; and grace period within which any policy-relevant flaws should be fixed.

Policy constraints

Policies have three main constraints that can be applied: rules, required scans, and remediation grace periods.

Evaluating applications against a policy

When an application is evaluated against a policy, it can receive one of four assessments:

Not assessed The application has not yet had a scan published
Passed The application has passed all the aspects of the policy, including rules, required scans, and grace period.
Did not pass The application has not completed all required scans; has not achieved the target Veracode Level; or has one or more policy relevant flaws that have exceeded the grace period to fix.
Conditional pass The application has one or more policy relevant flaws that have not yet exceeded the grace period to fix.

Understand Veracode Levels

The Veracode Level (VL) achieved by an application is determined by type of testing performed on the application, and the severity and types of flaws detected. A minimum security score (defined below) is also required for each level.

There are five Veracode Levels denoted as VL1, VL2, VL3, VL4, and VL5. VL1 is the lowest level and is achieved by demonstrating that security testing, automated static or dynamic, is utilized during the SDLC. VL5 is the highest level and is achieved by performing automated and manual testing and removing all significant flaws. The Veracode Levels VL2, VL3, and VL4 form a continuum of increasing software assurance between VL1 and VL5.

For IT staff operating applications, Veracode Levels can be used to set application security policies. For deployment scenarios of different business criticality, differing VLs should be made requirements. For example, the policy for applications that handle credit card transactions, and therefore have PCI compliance requirements, should be VL5. A medium business criticality internal application could have a policy requiring VL3.

Software developers can decide which VL they want to achieve based on the requirements of their customers. Developers of software that is mission critical to most of their customers will want to achieve VL5. Developers of general purpose business software may want

to achieve VL3 or VL4. Once the software has achieved a Veracode Level it can be communicated to customers through a Veracode Report or through the Veracode Directory on the Veracode web site.

Criteria for achieving Veracode Levels

The following table defines the details to achieve each Veracode Level. The criteria for all columns: Flaw Severities Not Allowed, Flaw Categories not Allowed, Testing Required, and Minimum Score.

*Dynamic is only an option for web applications.

Veracode Level	Flaw Severities Not Allowed	Testing Required*	Minimum Score
VL5	V.High, High, Medium	Static AND Manual	90
VL4	V.High, High, Medium	Static	80
VL3	V.High, High	Static	70
VL2	V.High	Static OR Dynamic OR Manual	60
VL1		Static OR Dynamic OR Manual	

When multiple testing techniques are used it is likely that not all testing will be performed on the exact same build. If that is the case the latest test results from a particular technique will be used to calculate the current Veracode Level. After 6 months test results will be deemed out of date and will no longer be used to calculate the current Veracode Level.

Business Criticality

The foundation of the Veracode rating system is the concept that more critical applications require higher security quality scores to be acceptable risks. Less business critical applications can tolerate lower security quality. The business criticality is dictated by the typical deployed environment and the value of data used by the application. Factors that determine business criticality are: reputation damage, financial loss, operational risk, sensitive information disclosure, personal safety, and legal violations.

US. Govt. OMB Memorandum M-04-04; NIST FIPS Pub. 199

Business Criticality	Description
Very High	Mission critical for business/safety of life and limb on the line
High	Exploitation causes serious brand damage and financial loss with long term business impact
Medium	Applications connected to the internet that process financial or private customer information
Low	Typically internal applications with non-critical business impact
Very Low	Applications with no material business impact

Business Criticality Definitions

Very High (BC5) This is typically an application where the safety of life or limb is dependent on the system; it is mission critical the application maintain 100% availability for the long term viability of the project or business. Examples are control software for industrial, transportation or medical equipment or critical business systems such as financial trading systems.
High (BC4) This is typically an important multi-user business application reachable from the internet and is critical that the application maintain high availability to accomplish its mission. Exploitation of high criticality applications cause serious brand damage and business/financial loss and could lead to long term business impact.
Medium (BC3) This is typically a multi-user application connected to the internet or any system that processes financial or private customer information. Exploitation of medium criticality applications typically result in material business impact resulting

in some financial loss, brand damage or business liability. An example is a financial services company's internal 401K management system.

Low (BC2) This is typically an internal only application that requires low levels of application security such as authentication to protect access to non-critical business information and prevent IT disruptions. Exploitation of low criticality applications may lead to minor levels of inconvenience, distress or IT disruption. An example internal system is a conference room reservation or business card order system.

Very Low (BC1) Applications that have no material business impact should its confidentiality, data integrity and availability be affected. Code security analysis is not required for applications at this business criticality, and security spending should be directed to other higher criticality applications.

Scoring Methodology

The Veracode scoring system, Security Quality Score, is built on the foundation of two industry standards, the Common Weakness Enumeration (CWE) and Common Vulnerability Scoring System (CVSS). CWE provides the dictionary of security flaws and CVSS provides the foundation for computing severity, based on the potential Confidentiality, Integrity and Availability impact of a flaw if exploited.

The Security Quality Score is a single score from 0 to 100, where 0 is the most insecure application and 100 is an application with no detectable security flaws. The score calculation includes non-linear factors so that, for instance, a single Severity 5 flaw is weighted more heavily than five Severity 1 flaws, and so that each additional flaw at a given severity contributes progressively less to the score.

Veracode assigns a severity level to each flaw type based on three foundational application security requirements — Confidentiality, Integrity and Availability. Each of the severity levels reflects the potential business impact if a security breach occurs across one or more of these security dimensions.

Confidentiality Impact

According to CVSS, this metric measures the impact on confidentiality if a exploit should occur using the vulnerability on the target system. At the weakness level, the scope of the Confidentiality in this model is within an application and is measured at three levels of impact -None, Partial and Complete.

Integrity Impact

This metric measures the potential impact on integrity of the application being analyzed. Integrity refers to the trustworthiness and guaranteed veracity of information within the application. Integrity measures are meant to protect data from unauthorized modification. When the integrity of a system is sound, it is fully proof from unauthorized modification of its contents.

Availability Impact

This metric measures the potential impact on availability if a successful exploit of the vulnerability is carried out on a target application. Availability refers to the accessibility of information resources. Almost exclusive to this domain are denial-of-service vulnerabilities. Attacks that compromise authentication and authorization for application access, application memory, and administrative privileges are examples of impact on the availability of an application.

Security Quality Score Calculation

The overall Security Quality Score is computed by aggregating impact levels of all weaknesses within an application and representing the score on a 100 point scale. This score does not predict vulnerability potential as much as it enumerates the security weaknesses and their impact levels within the application code.

The Raw Score formula puts weights on each flaw based on its impact level. These weights are exponential and determined by empirical analysis by Veracode's application security experts with validation from industry experts. The score is normalized to a scale of 0 to 100, where a score of 100 is an application with 0 detected flaws using the analysis technique for the application's business criticality.

Understand Severity, Exploitability, and Remediation Effort

Severity and exploitability are two different measures of the seriousness of a flaw. Severity is defined in terms of the potential impact to confidentiality, integrity, and availability of the application as defined in the CVSS, and exploitability is defined in terms of the likelihood

or ease with which a flaw can be exploited. A high severity flaw with a high likelihood of being exploited by an attacker is potentially more dangerous than a high severity flaw with a low likelihood of being exploited.

Remediation effort, also called Complexity of Fix, is a measure of the likely effort required to fix a flaw. Together with severity, the remediation effort is used to give Fix First guidance to the developer.

Veracode Flaw Severities

Veracode flaw severities are defined as follows:

Severity	Description
Very High	The offending line or lines of code is a very serious weakness and is an easy target for an attacker. The code should be modified immediately to avoid potential attacks.
High	The offending line or lines of code have significant weakness, and the code should be modified immediately to avoid potential attacks.
Medium	A weakness of average severity. These should be fixed in high assurance software. A fix for this weakness should be considered after fixing the very high and high for medium assurance software.
Low	This is a low priority weakness that will have a small impact on the security of the software. Fixing should be consideration for high assurance software. Medium and low assurance software can ignore these flaws.
Very Low	Minor problems that some high assurance software may want to be aware of. These flaws can be safely ignored in medium and low assurance software.
Informational	Issues that have no impact on the security quality of the application but which may be of interest to the reviewer.

Informational findings

Informational severity findings are items observed in the analysis of the application that have no impact on the security quality of the application but may be interesting to the reviewer for other reasons. These findings may include code quality issues, API usage, and other factors.

Informational severity findings have no impact on the security quality score of the application and are not included in the summary tables of flaws for the application.

Exploitability

Each flaw instance in a static scan may receive an exploitability rating. The rating is an indication of the intrinsic likelihood that the flaw may be exploited by an attacker. Veracode recommends that the exploitability rating be used to prioritize flaw remediation within a particular group of flaws with the same severity and difficulty of fix classification.

The possible exploitability ratings include:

Exploitability	Description
V. Unlikely	Very unlikely to be exploited
Unlikely	Unlikely to be exploited

Exploitability	Description
Neutral	Neither likely nor unlikely to be exploited.
Likely	Likely to be exploited
V. Likely	Very likely to be exploited

Note: All reported flaws found via dynamic scans are assumed to be exploitable, because the dynamic scan actually executes the attack in question and verifies that it is valid.

Effort/Complexity of Fix

Each flaw instance receives an effort/complexity of fix rating based on the classification of the flaw. The effort/complexity of fix rating is given on a scale of 1 to 5, as follows:

Effort/Complexity of Fix	Description
5	Complex design error. Requires significant redesign.
4	Simple design error. Requires redesign and up to 5 days to fix.
3	Complex implementation error. Fix is approx. 51-500 lines of code. Up to 5 days to fix.
2	Implementation error. Fix is approx. 6-50 lines of code. 1 day to fix.
1	Trivial implementation error. Fix is up to 5 lines of code. One hour or less to fix.

Flaw Types by Severity Level

The flaw types by severity level table provides a summary of flaws found in the application by Severity and Category. The table puts the Security Quality Score into context by showing the specific breakout of flaws by severity, used to compute the score as described above. If multiple analysis techniques are used, the table includes a breakout of all flaws by category and severity for each analysis type performed.

Flaws by Severity

The flaws by severity chart shows the distribution of flaws by severity. An application can get a mediocre security rating by having a few high risk flaws or many medium risk flaws.

Flaws in Common Modules

The flaws in common modules listing shows a summary of flaws in shared dependency modules in this application. A shared dependency is a dependency that is used by more than one analyzed module. Each module is listed with the number of executables that consume it as a dependency and a summary of the impact on the application's security score of the flaws found in the dependency.

The score impact represents the amount that the application score would increase if all the flaws in the shared dependency module were fixed. This information can be used to focus remediation efforts on common modules with a higher impact on the application security score.

Only common modules that were uploaded with debug information are included in the Flaws in Common Modules listing.

Action Items

The Action Items section of the report provides guidance on the steps required to bring the application to a state where it passes its assigned policy. These steps may include fixing or mitigating flaws or performing additional scans. The section also includes best practice recommendations to improve the security quality of the application.

Common Weakness Enumeration (CWE)

The Common Weakness Enumeration (CWE) is an industry standard classification of types of software weaknesses, or flaws, that can lead to security problems. CWE is widely used to provide a standard taxonomy of software errors. Every flaw in a Veracode report is classified according to a standard CWE identifier.

More guidance and background about the CWE is available at http://cwe.mitre.org/data/index.html.

About Manual Assessments

The Veracode platform can include the results from a manual assessment (usually a penetration test or code review) as part of a report. These results differ from the results of automated scans in several important ways, including objectives, attack vectors, and common attack patterns.

A manual penetration assessment is conducted to observe the application code in a run-time environment and to simulate real-world attack scenarios. Manual testing is able to identify design flaws, evaluate environmental conditions, compound multiple lower risk flaws into higher risk vulnerabilities, and determine if identified flaws affect the confidentiality, integrity, or availability of the application.

Objectives

The stated objectives of a manual penetration assessment are:

- Perform testing, using proprietary and/or public tools, to determine whether it is possible for an attacker to:
- Circumvent authentication and authorization mechanisms
- Escalate application user privileges
- Hijack accounts belonging to other users
- Violate access controls placed by the site administrator
- Alter data or data presentation
- Corrupt application and data integrity, functionality and performance
- Circumvent application business logic
- Circumvent application session management
- Break or analyze use of cryptography within user accessible components
- Determine possible extent access or impact to the system by attempting to exploit vulnerabilities
- Score vulnerabilities using the Common Vulnerability Scoring System (CVSS)
- Provide tactical recommendations to address security issues of immediate consequence

Provide strategic recommendations to enhance security by leveraging industry best practices

Attack vectors

In order to achieve the stated objectives, the following tests are performed as part of the manual penetration assessment, when applicable to the platforms and technologies in use:

- Cross Site Scripting (XSS)
- SQL Injection
- Command Injection
- Cross Site Request Forgery (CSRF)
- Authentication/Authorization Bypass
- Session Management testing, e.g. token analysis, session expiration, and logout effectiveness
- Account Management testing, e.g. password strength, password reset, account lockout, etc.
- Directory Traversal
- Response Splitting
- Stack/Heap Overflows
- Format String Attacks

- Cookie Analysis
- Server Side Includes Injection
- Remote File Inclusion
- LDAP Injection
- XPATH Injection
- Internationalization attacks
- Denial of Service testing at the application layer only
- AJAX Endpoint Analysis
- Web Services Endpoint Analysis
- HTTP Method Analysis
- SSL Certificate and Cipher Strength Analysis
- Forced Browsing

CAPEC Attack Pattern Classification

The following attack pattern classifications are used to group similar application flaws discovered during manual penetration testing. Attack patterns describe the general methods employed to access and exploit the specific weaknesses that exist within an application. CAPEC (Common Attack Pattern Enumeration and Classification) is an effort led by Cigital, Inc. and is sponsored by the United States Department of Homeland Security's National Cyber Security Division.

Abuse of Functionality

Exploitation of business logic errors or misappropriation of programmatic resources. Application functions are developed to specifications with particular intentions, and these types of attacks serve to undermine those intentions.

Examples:

- Exploiting password recovery mechanisms
- Accessing unpublished or test APIs
- Cache poisoning

Spoofing

Impersonation of entities or trusted resources. A successful attack will present itself to a verifying entity with an acceptable level of authenticity.

Examples:

- Man in the middle attacks
- Checksum spoofing
- Phishing attacks

Probabilistic Techniques

Using predictive capabilities or exhaustive search techniques in order to derive or manipulate sensitive information. Attacks capitalize on the availability of computing resources or the lack of entropy within targeted components.

Examples:

- Password brute forcing
- Cryptanalysis
- Manipulation of authentication tokens

Exploitation of Authentication

Circumventing authentication requirements to access protected resources. Design or implementation flaws may allow authentication checks to be ignored, delegated, or bypassed.

Examples:

- Cross-site request forgery
- Reuse of session identifiers
- Flawed authentication protocol

Resource Depletion

Affecting the availability of application components or resources through symmetric or asymmetric consumption. Unrestricted access to computationally expensive functions or implementation flaws that affect the stability of the application can be targeted by an attacker in order to cause denial of service conditions.

Examples:

- Flooding attacks
- Unlimited file upload size
- Memory leaks

Exploitation of Privilege/Trust

Undermining the application's trust model in order to gain access to protected resources or gain additional levels of access as defined by the application. Applications that implicitly extend trust to resources or entities outside of their direct control are susceptible to attack.

Examples:

- Insufficient access control lists
- Circumvention of client side protections
- Manipulation of role identification information

Injection

Inserting unexpected inputs to manipulate control flow or alter normal business processing. Applications must contain sufficient data validation checks in order to sanitize tainted data and prevent malicious, external control over internal processing.

Examples:

- SQL Injection
- Cross-site scripting
- XML Injection

Data Structure Attacks

Supplying unexpected or excessive data that results in more data being written to a buffer than it is capable of holding. Successful attacks of this class can result in arbitrary command execution or denial of service conditions.

Examples:

- Buffer overflow
- Integer overflow
- Format string overflow

Data Leakage Attacks

Recovering information exposed by the application that may itself be confidential or may be useful to an attacker in discovering or exploiting other weaknesses. A successful attack may be conducted passive observation or active interception methods. This attack pattern often manifests itself in the form of applications that expose sensitive information within error messages.

Examples:

- Sniffing clear-text communication protocols
- Stack traces returned to end users
- Sensitive information in HTML comments

Resource Manipulation

Manipulating application dependencies or accessed resources in order to undermine security controls and gain unauthorized access to protected resources. Applications may use tainted data when constructing paths to local resources or when constructing processing environments.

Examples:

- Carriage Return Line Feed log file injection
- File retrieval via path manipulation
- User specification of configuration files

Time and State Attacks

Undermining state condition assumptions made by the application or capitalizing on time delays between security checks and performed operations. An application that does not enforce a required processing sequence or does not handle concurrency adequately will be susceptible to these attack patterns.

Examples:

- Bypassing intermediate form processing steps
- Time-of-check and time-of-use race conditions
- Deadlock triggering to cause a denial of service

Terms of Use

Use and distribution of this report are governed by the agreement between Veracode and its customer. In particular, this report and the results in the report cannot be used publicly in connection with Veracode's name without written permission.

Appendix A: Approved Mitigated Flaws (by Collegesource, Inc.)

NOTE: Except in circumstances where the Customer has purchased Veracode's Mitigation Proposal Review Solution, Veracode does not review the mitigation strategy described below and is not responsible for its contents or the accuracy of any statements provided.

High (3 flaws)

SQL Injection(3 flaws)

Associated Flaws by CWE ID:

Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') (CWE ID 89)(3 flaws)

Fix Required by Policy: Flaw no longer impacts results.
Flaw continues to impact results.

Instances found via Static Scan

Flaw Id	Module	Exploitability	Mitigation Comment
436	Dashboard.war	V.Likely	*Potential False Positive (Collegesource, Inc.):*We are properly passing in parameters to the jdbc template and using a prepared statement *Approve Mitigation (Collegesource, Inc.):*These have been reviewd in a previsous release
373	SelfService.war/redLanternActionService-4.5.4.2.jar	V.Likely	*Mitigate by Design (Collegesource, Inc.):*These queries are ran using our criteria query building tool that limits what a user can input *Approve Mitigation (Collegesource, Inc.):*These have been reviewd in a previsous release *Reject Mitigation (Collegesource, Inc.):*-needs more discussion *Mitigate by Design (Collegesource, Inc.):*The values being appended to the queries were handled by issue UACH-6200. The remaining part of the query being append is just the operands which come from a static list. So they are not a risk for SQL Injection attacks. *Approve Mitigation (Collegesource, Inc.):*See UACH-6200.
379	SelfService.war/redLanternActionService-4.5.4.2.jar	V.Likely	*Mitigate by Design (Collegesource, Inc.):*These queries are ran using our criteria query building tool that limits what a user can input *Approve Mitigation (Collegesource, Inc.):*These have been reviewd in a previsous release *Reject Mitigation (Collegesource, Inc.):*-needs more discussion *Mitigate by Design (Collegesource, Inc.):*The values being appended to the queries were handled by issue UACH-6200. The remaining part of the query being append is just the operands which come from a static list. So they are not a risk

Flaw Id	Module	Exploitability	Mitigation Comment
			for SQL Injection attacks. *Approve Mitigation (Collegesource, Inc.):*See UACH-6200.

Medium (197 flaws)

Credentials Management(3 flaws)

Associated Flaws by CWE ID:

Use of Hard-coded Password (CWE ID 259)(3 flaws)

Fix Required by Policy: Flaw no longer impacts results. Flaw continues to impact results.

Instances found via Static Scan

Flaw Id	Module	Exploitability	Mitigation Comment
76	Schedmule.war/collegesource-security-api-4.5.4.2.jar	Likely	*Potential False Positive (Collegesource, Inc.):*We are storing password attributes in a variable in these classes rather than having an insecure, hardcoded password. *Approve Mitigation (Collegesource, Inc.):*Reviewed with 4.5.4.1 testing
216	Schedmule.war/redLanternActionService-4.5.4.2.jar	Likely	*Potential False Positive (Collegesource, Inc.):*We are storing password attributes in a variable in these classes rather than having an insecure, hardcoded password. *Approve Mitigation (Collegesource, Inc.):*Reviewed with 4.5.4.1 testing
280	Planner.war	Likely	*Potential False Positive (Collegesource, Inc.):*We are storing password attributes in a variable in these classes rather than having an insecure, hardcoded password. *Approve Mitigation (Collegesource, Inc.):*Reviewed with 4.5.4.1 testing

Cryptographic Issues(1 flaw)

Associated Flaws by CWE ID:

Insufficient Entropy (CWE ID 331)(1 flaw)

Instances found via Static Scan

Flaw Id	Module	Exploitability	Mitigation Comment
365	Planner.war	Unlikely	*Potential False Positive (Collegesource, Inc.):*UACH-6202

Flaw Id	Module	Exploitability	Mitigation Comment
			After reviewing the code this is a false positive we are just using math.random to generate a temporary id for the audit returned for the front end of planner it is never perssited or used or a user or session id. *Approve Mitigation (Collegesource, Inc.):*Reviewed with 4.5.4.1 testing

Session Fixation(4 flaws)

Associated Flaws by CWE ID:

Session Fixation (CWE ID 384)(4 flaws)

Fix Required by Policy: Flaw no longer impacts results.
Flaw continues to impact results.

Instances found via Static Scan

Flaw Id	Module	Exploitability	Mitigation Comment
1283	Schedmule.war	Neutral	*Potential False Positive (Collegesource, Inc.):*This flaw is a false positive we are just grabbing the session out the request to look at the cookie config for newly created custom cookies. We are not creating a new session or modifying the existing session that would create a risk for this type of attack. *Approve Mitigation (Collegesource, Inc.):*reviewed
1304	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*This flaw is a false positive we are just grabbing the session out the request to look at the cookie config for newly created custom cookies. We are not creating a new session or modifying the existing session that would create a risk for this type of attack. *Approve Mitigation (Collegesource, Inc.):*reviewed
1280	Dashboard.war	Neutral	*Potential False Positive (Collegesource, Inc.):*This flaw is a false positive we are just grabbing the session out the request to look at the cookie config for newly created custom cookies. We are not creating a new session or modifying the existing session that would create a risk for this type of attack. *Approve Mitigation (Collegesource, Inc.):*reviewed
960	Planner.war	Neutral	*Mitigate by Design (Collegesource, Inc.):*DASH-1038

Flaw Id	Module	Exploitability	Mitigation Comment
			This security issue seemed to be caused we were opening a session in the doFilter of the SessionTimeoutCookieFitler which is outside the scope of the authenticated session. But we were only opening it to look at the max interval time. So instead of assigning the session to a variable, I changed the code to just pull the session and get the max interval time with declaring a session variable to remove the risk of Session Fixation *Approve Mitigation (Collegesource, Inc.):*Reviewed with 4.5.4.1 testing

Cross-Site Scripting (XSS)(17 flaws)

Associated Flaws by CWE ID:

Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) (CWE ID 80)(17 flaws)

Fix Required by Policy: Flaw no longer impacts results. Flaw continues to impact results.

Instances found via Static Scan

Flaw Id	Module	Exploitability	Mitigation Comment
170	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed *Approve Mitigation (Collegesource, Inc.):*See UACH-6200.
217	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed *Approve Mitigation (Collegesource, Inc.):*See UACH-6200.
186	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed
394	SelfService.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed
168	SelfService.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are using

Flaw Id	Module	Exploitability	Mitigation Comment
			antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed *Approve Mitigation (Collegesource, Inc.):*See UACH-6200.
223	SelfService.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed
180	SelfService.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed
153	SelfService.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed *Approve Mitigation (Collegesource, Inc.):*See UACH-6200.
152	SelfService.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed
148	SelfService.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed
136	SelfService.war	Neutral	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed
740	Schedmule.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are currently using antisamy to encode any data that is submitted to the server. So any HTML characters including '<', '>', and '&' will be encoded before being sent back to the client browser. *Approve Mitigation (Collegesource, Inc.):*reviewed
197	SelfService.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed
377	SelfService.war	V.Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to scrub any user input used in the returned json to

Flaw Id	Module	Exploitability	Mitigation Comment
			prevent XSS. *Approve Mitigation (Collegesource, Inc.):*-reviewed
973	Schedmule.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are currently using antisamy to encode any data that is submitted to the server. So any HTML characters including '<', '>', and '&' will be encoded before being sent back to the client browser. *Approve Mitigation (Collegesource, Inc.):*reviewed
434	Schedmule.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are currently using antisamy to encode any data that is submitted to the server. So any HTML characters including '<', '>', and '&' will be encoded before being sent back to the client browser. *Approve Mitigation (Collegesource, Inc.):*reviewed
1325	Schedmule.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are currently using antisamy to encode any data that is submitted to the server. So any HTML characters including '<', '>', and '&' will be encoded before being sent back to the client browser. *Approve Mitigation (Collegesource, Inc.):*reviewed

CRLF Injection(172 flaws)

Associated Flaws by CWE ID:

Improper Neutralization of CRLF Sequences ('CRLF Injection') (CWE ID 93)(10 flaws)

Fix Required by Policy: Flaw no longer impacts results. Flaw continues to impact results.

Instances found via Static Scan

Flaw Id	Module	Exploitability	Mitigation Comment
218	Planner.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
301	Planner.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
326	Planner.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
143	Schedmule.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
334	Planner.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
203	Planner.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
274	Schedmule.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
345	Planner.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue

Flaw Id	Module	Exploitability	Mitigation Comment
			bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
350	Planner.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
193	Schedmule.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed

Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Response Splitting') (CWE ID 113)(5 flaws)

Fix Required by Policy: Flaw no longer impacts results. Flaw continues to impact results.

Instances found via Static Scan

Flaw Id	Module	Exploitability	Mitigation Comment
384	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using antisamy to scrub user input to prevent response splitting *Approve Mitigation (Collegesource, Inc.)*:-reviewed
390	SelfService.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using antisamy to scrub user input to prevent response splitting *Approve Mitigation (Collegesource, Inc.)*:-reviewed *Approve Mitigation (Collegesource, Inc.)*:See UACH-6200.
1314	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using antisamy to protect against special characters coming in through the request that could be used to modify response headers. *Approve Mitigation (Collegesource, Inc.)*:reviewed
1312	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using

Flaw Id	Module	Exploitability	Mitigation Comment
			antisamy to protect against special characters coming in through the request that could be used to modify response headers. *Approve Mitigation (Collegesource, Inc.):*reviewed
1279	Dashboard.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using antisamy to protect against special characters coming in through the request that could be used to modify response headers. *Approve Mitigation (Collegesource, Inc.):*reviewed

Improper Output Neutralization for Logs (CWE ID 117)(157 flaws)

Fix Required by Policy: Flaw no longer impacts results. Flaw continues to impact results.

Instances found via Static Scan

Flaw Id	Module	Exploitability	Mitigation Comment
352	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
351	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
282	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
269	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935

Flaw Id	Module	Exploitability	Mitigation Comment
			Approve Mitigation (Collegesource, Inc.):-reviewed
354	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
307	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
75	Dashboard.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2

Flaw Id	Module	Exploitability	Mitigation Comment
			and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
409	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
412	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue

Flaw Id	Module	Exploitability	Mitigation Comment
			bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
258	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
299	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
343	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
331	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
425	Dashboard.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935

Flaw Id	Module	Exploitability	Mitigation Comment
			*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
427	Dashboard.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
426	Dashboard.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being

Flaw Id	Module	Exploitability	Mitigation Comment
			written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
428	Dashboard.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
424	Dashboard.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
976	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
179	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1303	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
1311	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
1300	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue

Flaw Id	Module	Exploitability	Mitigation Comment
			bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
130	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
116	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
122	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
119	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
213	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
158	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2

Flaw Id	Module	Exploitability	Mitigation Comment
			and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
178	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
140	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
188	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
181	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
156	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935

Flaw Id	Module	Exploitability	Mitigation Comment
			Approve Mitigation (Collegesource, Inc.):-reviewed
214	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
137	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
162	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
183	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
233	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
393	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
106	Schedmule.war/collegesource-security-api-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
102	Schedmule.war/collegesource-security-api-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
59	Dashboard.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
303	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
190	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1310	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
120	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1302	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
803	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
807	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
800	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
805	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
812	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
801	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being

Flaw Id	Module	Exploitability	Mitigation Comment
			written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
284	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
276	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
252	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
340	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
336	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
271	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
296	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
308	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935

Flaw Id	Module	Exploitability	Mitigation Comment
			*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
357	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
338	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being

Flaw Id	Module	Exploitability	Mitigation Comment
			written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
321	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
121	Schedmule.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
149	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue

Flaw Id	Module	Exploitability	Mitigation Comment
			bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
160	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
263	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1322	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
253	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1301	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
124	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2

Flaw Id	Module	Exploitability	Mitigation Comment
			and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
341	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
808	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
814	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
806	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2

Flaw Id	Module	Exploitability	Mitigation Comment
			and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
815	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
809	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
191	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
199	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1072	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935

Flaw Id	Module	Exploitability	Mitigation Comment
			*Approve Mitigation (Collegesource, Inc.):*reviewed
305	Planner.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
291	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1319	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2

Flaw Id	Module	Exploitability	Mitigation Comment
			and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
1315	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
55	Schedmule.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
278	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
802	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2

Flaw Id	Module	Exploitability	Mitigation Comment
			and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
262	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
285	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
318	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being

Flaw Id	Module	Exploitability	Mitigation Comment
			written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
279	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1321	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
1317	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue

Flaw Id	Module	Exploitability	Mitigation Comment
			bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
342	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
293	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
332	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
294	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed

Flaw Id	Module	Exploitability	Mitigation Comment
323	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
283	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
315	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being

Flaw Id	Module	Exploitability	Mitigation Comment
			written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
962	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
359	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
328	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
333	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935

Flaw Id	Module	Exploitability	Mitigation Comment
			Approve Mitigation (Collegesource, Inc.):-reviewed
361	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
297	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
255	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2

Flaw Id	Module	Exploitability	Mitigation Comment
			and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
266	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
381	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
385	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being

Flaw Id	Module	Exploitability	Mitigation Comment
			written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
375	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
395	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
840	Schedmule.war/uachieve-apis-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
382	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
376	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
145	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
423	Dashboard.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed

Flaw Id	Module	Exploitability	Mitigation Comment
1022	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
1308	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
344	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
804	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed

Flaw Id	Module	Exploitability	Mitigation Comment
813	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
337	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
317	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being

Flaw Id	Module	Exploitability	Mitigation Comment
			written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
265	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
360	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
330	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue

Flaw Id	Module	Exploitability	Mitigation Comment
			bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
277	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
164	Schedmule.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
224	Schedmule.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
275	Planner.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1116	SelfService.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
1127	SelfService.war/redLanternActionService-4.5.4.2.jar	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
1137	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
1307	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
1306	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
221	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:-reviewed
1305	SelfService.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:reviewed
261	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.)*:-reviewed
312	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.)*:We are using log2j and adding the built-in HTML encoding to all messages being

Flaw Id	Module	Exploitability	Mitigation Comment
			written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
353	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
363	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
288	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
1318	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935

Flaw Id	Module	Exploitability	Mitigation Comment
			*Approve Mitigation (Collegesource, Inc.):*reviewed
1323	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*reviewed
309	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
368	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2

Flaw Id	Module	Exploitability	Mitigation Comment
			and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
356	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
370	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue

Flaw Id	Module	Exploitability	Mitigation Comment
			bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
366	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
316	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow:

Flaw Id	Module	Exploitability	Mitigation Comment
			DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
302	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
367	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed

Flaw Id	Module	Exploitability	Mitigation Comment
314	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log2j and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed
254	Planner.war	Likely	*Mitigate by Design (Collegesource, Inc.):*We are using log4j2 and adding the built-in HTML encoding to all messages being written to the log files and console. See the related Jira issue bellow: DASH-935 *Approve Mitigation (Collegesource, Inc.):*-reviewed

Low (10 flaws)

Information Leakage(10 flaws)

Associated Flaws by CWE ID:

Generation of Error Message Containing Sensitive Information (CWE ID 209)(10 flaws)

Fix Required by Policy: Flaw no longer impacts results. Flaw continues to impact results.

Instances found via Static Scan

Flaw Id	Module	Exploitability	Mitigation Comment
211	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are

Flaw Id	Module	Exploitability	Mitigation Comment
			making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed
196	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed
226	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed
383	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed
198	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed
155	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed
202	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed

Flaw Id	Module	Exploitability	Mitigation Comment
146	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed
195	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed
212	SelfService.war	Neutral	*Potential False Positive (Collegesource, Inc.):*These are falase positives it is marking the audit data that we are making into a josn object as sensitive data exposure but we are purposely adding that data to the page for building the marker tree *Approve Mitigation (Collegesource, Inc.):*reviewed