Veracode Detailed Report

Application Security Report
As of 7 Dec 2021

Prepared for:	Visionet
Prepared on:	December 8, 2021
Application:	PNC Indexing
Sandbox:	Development Sandbox
Industry:	Financial Services
Business Criticality:	BC5 (Very High)
Required Analysis:	Static
Type(s) of Analysis Conducted:	Static
Scope of Static Scan:	1 of 87 Modules Analyzed

Inside This Report

While every precaution has been taken in the preparation of this document, Veracode, Inc. assumes no responsibility for errors, omissions, or for damages resulting from the use of the information herein. The Veracode platform uses static and/or dynamic analysis techniques to discover potentially exploitable flaws. Due to the nature of software security testing, the lack of discoverable flaws does not mean the software is 100% secure.

© 2021 Veracode, Inc.

65 Network Drive, Burlington, MA 01803

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

Veracode Detailed Report

Application Security Report
As of 7 Dec 2021

Veracode Level: VL3 + SCA

Rated: Dec 7, 2021

Application:	PNC Indexing	Business Criticality:	Very High
Target Level:	VL3 + SCA	Published Rating:	C

Scans Included in Report

Static Scan	Dynamic Scan	Manual Penetration Test
7 Dec 2021 Static (2) Score: 76 Completed: 12/7/21	Not Included in Report	Not Included in Report

Executive Summary

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.

Application Business Criticality: BC5 (Very High)

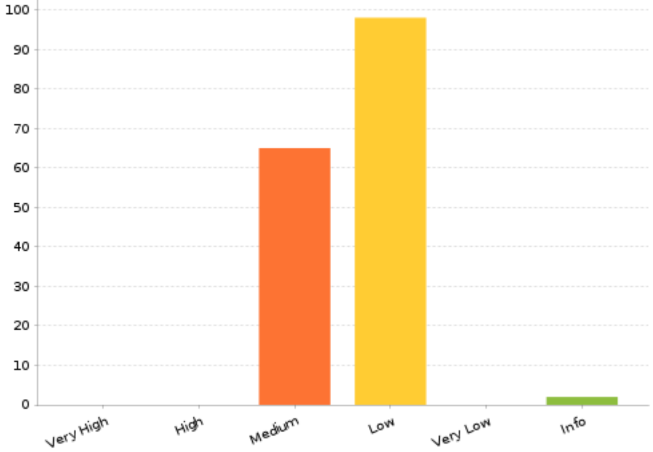
Impacts:Operational Risk (High), Financial Loss (High)

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.

Analyses Performed vs. Required

	Any	Static	Dynamic	Manual Penetration Test
Performed:		●	○	○
Required:	○	●	○	○

Summary of Flaws Found by Severity



Action Items:

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.

Required Analysis

→ Your policy requires periodic Static Scan. Your next analysis must be completed by 3/7/22. Please submit your application for Static Scan by the deadline and remediate the required detected flaws to conform to your assigned policy.

Flaw Severities

→ High severity flaws and above must be fixed for policy compliance.

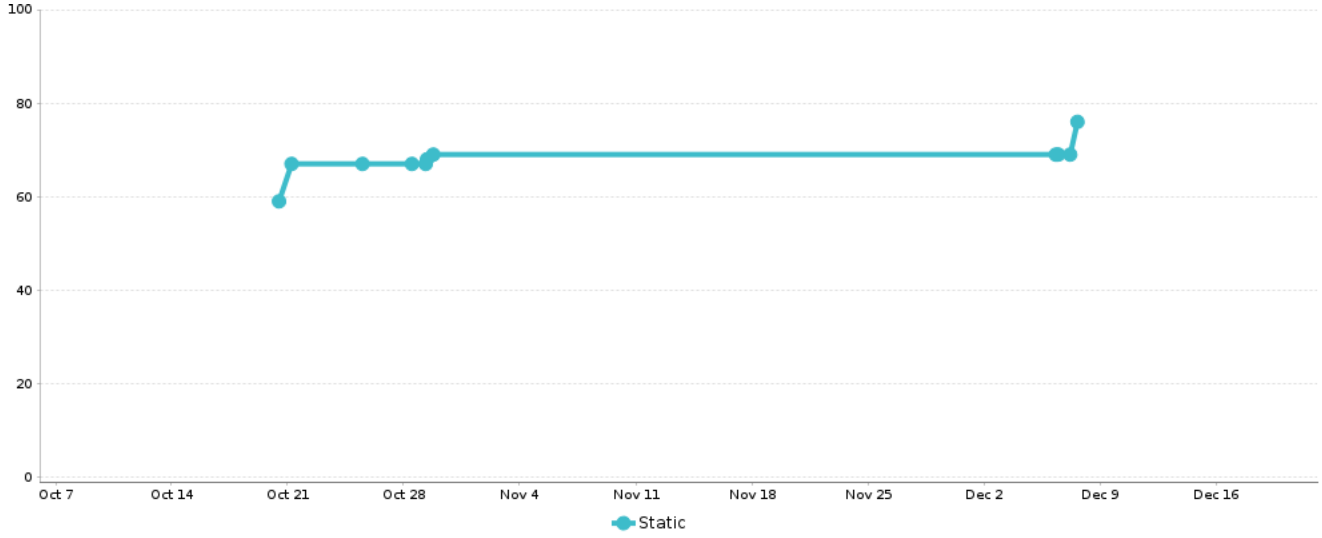
© 2021 Veracode, Inc.

Longer Timeframe (6 - 12 months)

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

© 2021 Veracode, Inc.

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: 20211110154823

The following modules were included in the application scan:

Module Name	Compiler	Operating Environment	Engine Version
Visionet.VLR.Web.Presentation.dll	MSIL_MSVC14_X86	Windows	20211110154823

File Differences Between Scans

The uploaded modules for this scan do not match the modules you uploaded for the previous scan. This disparity can affect the scan results even if Veracode did not find flaws in the files with differences. See appendix for more details.

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
AntiXSSLibrary.dll	MSIL_MSVC11_X86	Windows	20211110154823
Antlr3.Runtime.dll	MSIL_MSVC11_X86	Windows	20211110154823
Aspose.Cells.dll	MSIL_MSVC6	Windows	20211110154823
Aspose.Words.dll	MSIL_MSVC8_X86	Windows	20211110154823
BGC128.dll	MSIL_MSVC6	Windows	20211110154823
CrystalDecisions.CrystalReports.Engine.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.ClientDoc.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.CommLayer.dll	MSIL_MSVC8_X86	Windows	20211110154823

Module Name	Compiler	Operating Environment	Engine Version
CrystalDecisions.ReportAppServer.CommonControls.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.CommonObjectModel.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.Controllers.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.CubeDefModel.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.DataDefModel.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.DataSetConversion.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.ObjectFactory.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.Prompting.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.ReportDefModel.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.ReportAppServer.XmlSerialize.dll	MSIL_MSVC8_X86	Windows	20211110154823
CrystalDecisions.Shared.dll	MSIL_MSVC8_X86	Windows	20211110154823
EPPlus.dll	MSIL_MSVC14_X86	Windows	20211110154823
HtmlSanitizationLibrary.dll	MSIL_MSVC8_X86	Windows	20211110154823
Interop.Microsoft.Office.Core.dll	MSIL_MSVC6	Windows	20211110154823
Interop.VBIDE.dll	MSIL_MSVC6	Windows	20211110154823
Interop.Word.dll	MSIL_MSVC6	Windows	20211110154823
Ionic.Zip.dll	MSIL_MSVC8_X86	Windows	20211110154823
itextsharp.dll	MSIL_MSVC11_X86	Windows	20211110154823
JS files within Packages.zip	JAVASCRIPT_5_1	JavaScript	20211110154823
log4net.dll	MSIL_MSVC6	Windows	20211110154823
Microsoft.Office.Interop.Word.dll	MSIL_MSVC6	Windows	20211110154823
Microsoft.Practices.EnterpriseLibrary.Common.dll	MSIL_MSVC6	Windows	20211110154823
Microsoft.Practices.EnterpriseLibrary.Configuration.dll	MSIL_MSVC6	Windows	20211110154823
Microsoft.Practices.EnterpriseLibrary.Data.dll	MSIL_MSVC6	Windows	20211110154823
Microsoft.Practices.EnterpriseLibrary.ExceptionHandling.dll	MSIL_MSVC6	Windows	20211110154823

Module Name	Compiler	Operating Environment	Engine Version
Microsoft.Practices.EnterpriseLibrary.Logging.dll	MSIL_MSVC6	Windows	20211110154823
Microsoft.Vbe.Interop.dll	MSIL_MSVC6	Windows	20211110154823
Microsoft.Web.Infrastructure.dll	MSIL_MSVC8_X86	Windows	20211110154823
Microsoft.Web.UI.WebControls.dll	MSIL_MSVC6	Windows	20211110154823
Nest.dll	MSIL_MSVC11_X86	Windows	20211110154823
Newtonsoft.Json.dll	MSIL_MSVC11_X86	Windows	20211110154823
office.dll	MSIL_MSVC6	Windows	20211110154823
PdfSharp.dll	MSIL_MSVC14_X86	Windows	20211110154823
Recaptcha.dll	MSIL_MSVC8_X86	Windows	20211110154823
RestClient.Net.dll	MSIL_MSVC14_X86	Windows	20211110154823
stdole.dll	MSIL_MSVC6	Windows	20211110154823
System.Data.Entity.dll	MSIL_MSVC11_X86	Windows	20211110154823
System.Web.Helpers.dll	MSIL_MSVC14_X86	Windows	20211110154823
System.Web.Mvc.dll	MSIL_MSVC14_X86	Windows	20211110154823
System.Web.Optimization.dll	MSIL_MSVC11_X86	Windows	20211110154823
System.Web.Razor.dll	MSIL_MSVC14_X86	Windows	20211110154823
System.Web.WebPages.Deployment.dll	MSIL_MSVC14_X86	Windows	20211110154823
System.Web.WebPages.dll	MSIL_MSVC14_X86	Windows	20211110154823
System.Web.WebPages.Razor.dll	MSIL_MSVC14_X86	Windows	20211110154823
TwoFactorAuth.Net.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Bre.DAL.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Bre.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Bre.Interface.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Correspondence.Common.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Correspondence.DAL.dll	MSIL_MSVC14_X86	Windows	20211110154823

Module Name	Compiler	Operating Environment	Engine Version
Visionet.Correspondence.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.EMail.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Facade.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Facade.XmlSerializers.dll	MSIL_MSVC11_X86	Windows	20211110154823
Visionet.LenderQB.OCRVendor.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Logging.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Visiflow.BLL.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Visiflow.Common.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.Visiflow.DAL.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.Backend.Facade.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.Common.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.DomainRepository.Common.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.DomainRepository.LoanReview.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.DomainRepository.OCROrder.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.DomainRepository.ScreenConfigurations.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.ElasticSearch.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.Model.dll	MSIL_MSVC14_X86_64	Windows X86_64	20211110154823
Visionet.VLR.OCRDataMining.Backend.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.OCRFileOrder.Backend.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.Web.Infrastructure.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.VLR.Web.Presentation.Utilities.dll	MSIL_MSVC14_X86	Windows	20211110154823
Visionnet.VLR.LabelPrinting.dll	MSIL_MSVC14_X86	Windows	20211110154823
VLR.ClassLibrary.dll	MSIL_MSVC14_X86	Windows	20211110154823
VLR.ClassLibrary.XmlSerializers.dll	MSIL_MSVC11_X86	Windows	20211110154823
VLR.Common.DAL.dll	MSIL_MSVC14_X86	Windows	20211110154823

Module Name	Compiler	Operating Environment	Engine Version
VLR.Common.dll	MSIL_MSVC14_X86	Windows	20211110154823
VLRCache.dll	MSIL_MSVC14_X86	Windows	20211110154823
WebGrease.dll	MSIL_MSVC8_X86	Windows	20211110154823

Flaw Types by Severity and Category

	Static Scan Security Quality Score = 76 (+7) from prior scan		
Very High	**0**		
High	**0**		
Medium	**65 (-52)**		
Cross-Site Scripting (XSS)	4		
Directory Traversal	58 (-50)		
Information Leakage	1 (-1)		
Insufficient Input Validation	2		
Time and State	(-1)		
Low	**98 (-1)**		
Cryptographic Issues	6 (-1)		
Information Leakage	34		
Insufficient Input Validation	58		
Very Low	**0**		
Informational	**2 (-5)**		
Code Quality	2 (-5)		
Total	**165 (-58)**		

Policy Evaluation

Policy Name: Veracode Recommended Medium + SCA

Revision: 1

Policy Status: Not Assessed

Description: Veracode provides default policies to make it easier for organizations to begin measuring their applications against policies. Veracode Recommended Policies are available for customers as an option when they are ready to move beyond the initial bar set by the Veracode Transitional Policies. The policies are based on the Veracode Level definitions.

Rules

Rule type	Requirement	Findings	Status
Minimum Veracode Level	VL3 + SCA	VL3 + SCA	Passed
(VL3 + SCA) Min Analysis Score	70	76	Passed
(VL3 + SCA) Max Severity	High	Flaws found: 0	Passed

Software Composition Analysis Rules

Rule type	Requirement	Findings	Status
(VL3 + SCA) Disallow Vulnerabilities by Severity	High and Above Not Allowed	0 Components	Passed
(VL3 + SCA) Disallow Component Blocklist	Prevent an application from passing policy if blocklisted components are detected	0 Blocklisted	Passed

Unsupported Frameworks

This report may have incomplete results based on the following unsupported frameworks identified during the static scan:

* BusinessObjects
* Crystal Reports

The lack of support for all frameworks in use by this application and/or its supporting libraries may prevent the static discovery of some flaws in the application, however, it does not invalidate the flaws that were found.

Findings & Recommendations

Best Practice Findings

Best Practices in use Throughout the Application

● This application correctly uses cryptographically secure random number generation.

Additionally, you are doing a good job at protecting against these flaw types:

● **Cross-Site Scripting (XSS)**
CWE–80: Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)

* This application has 6 opportunities for this flaw, and 2 were successfully defended against using security best practices.

* The remaining 4 flaws should be addressed and are described in the following section, “Detailed Flaws by Severity.”

Detailed Flaws by Severity

Very High (0 flaws)

No flaws of this type were found

High (0 flaws)

No flaws of this type were found

Medium (65 flaws)

→ Cross-Site Scripting (XSS)(4 flaws)

Description

Cross-site scripting (XSS) attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed occur whenever a web application uses untrusted data in the output it generates without validating or encoding it. XSS vulnerabilities are commonly exploited to steal or manipulate cookies, modify presentation of content, and compromise sensitive information, with new attack vectors being discovered on a regular basis. XSS is also commonly referred to as HTML injection.

XSS vulnerabilities can be either persistent or transient (often referred to as stored and reflected, respectively). In a persistent XSS vulnerability, the injected code is stored by the application, for example within a blog comment or message board. The attack occurs whenever a victim views the page containing the malicious script. In a transient XSS vulnerability, the injected code is included directly in the HTTP request. These attacks are often carried out via malicious URLs sent via email or another website and requires the victim to browse to that link. The consequence of an XSS attack to a victim is the same regardless of whether it is persistent or transient; however, persistent XSS vulnerabilities are likely to affect a greater number of victims due to its delivery mechanism.

Recommendations

Several techniques can be used to prevent XSS attacks. These techniques complement each other and address security at different points in the application. Using multiple techniques provides defense-in-depth and minimizes the likelihood of a XSS vulnerability.

- * Use output filtering to sanitize all output generated from user-supplied input, selecting the appropriate method of encoding based on the use case of the untrusted data. For example, if the data is being written to the body of an HTML page, use HTML entity encoding. However, if the data is being used to construct generated Javascript or if it is consumed by client-side methods that may interpret it as code (a common technique in Web 2.0 applications), additional restrictions may be necessary beyond simple HTML encoding.
- * Validate user-supplied input using positive filters (white lists) to ensure that it conforms to the expected format, using centralized data validation routines when possible.
- * Do not permit users to include HTML content in posts, notes, or other data that will be displayed by the application. If users are permitted to include HTML tags, then carefully limit access to specific elements or attributes, and use strict validation filters to prevent abuse.

Associated Flaws by CWE ID:

→ Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) (CWE ID 80)(4 flaws)

Description

This call contains a cross-site scripting (XSS) flaw. The application populates the HTTP response with untrusted input, allowing an attacker to embed malicious content, such as Javascript code, which will be executed in the context of the victim's browser. XSS vulnerabilities are commonly exploited to steal or manipulate cookies, modify presentation of content, and compromise confidential information, with new attack vectors being discovered on a regular basis.

Effort to Fix: 3 - Complex implementation error. Fix is approx. 51-500 lines of code. Up to 5 days to fix.

Recommendations

Use contextual escaping on all untrusted data before using it to construct any portion of an HTTP response. The escaping method should be chosen based on the specific use case of the untrusted data, otherwise it may not protect fully against the attack. For example, if the data is being written to the body of an HTML page, use HTML entity escaping; if the data is being written to an attribute, use attribute escaping; etc. When a web framework provides built-in support for automatic XSS escaping, do not disable it. Both the OWASP Java Encoder library for Java and the Microsoft AntiXSS library provide contextual escaping methods. For more details on contextual escaping, see https://www.owasp.org/index.php/XSS_%%28Cross_Site_Scripting%%29_Prevention_Cheat_Sheet. In addition, as a best practice, always validate untrusted input to ensure that it conforms to the expected format, using centralized data validation routines when possible.

Instances found via Static Scan

	Flaw Id	Module #	Class #	Module	Location	Fix By
	2397	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 3661	
	2388	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 3666	
NEW	2433	16	-	visionet.vlr.web.presentation.dll	.../loantaskcontroller.cs 1746	
NEW	2432	16	-	visionet.vlr.web.presentation.dll	.../loantaskcontroller.cs 1751	

Directory Traversal(58 flaws)

Description

Allowing user input to control paths used in filesystem operations may enable an attacker to access or modify otherwise protected system resources that would normally be inaccessible to end users. In some cases, the user-provided input may be passed directly to the filesystem operation, or it may be concatenated to one or more fixed strings to construct a fully-qualified path.

When an application improperly cleanses special character sequences in user-supplied filenames, a path traversal (or directory traversal) vulnerability may occur. For example, an attacker could specify a filename such as "../../etc/passwd", which resolves to a file outside of the intended directory that the attacker would not normally be authorized to view.

Recommendations

Assume all user-supplied input is malicious. Validate all user-supplied input to ensure that it conforms to the expected format, using centralized data validation routines when possible. When using black lists, be sure that the sanitizing routine performs a sufficient number of iterations to remove all instances of disallowed characters and ensure that the end result is not dangerous.

Associated Flaws by CWE ID:

External Control of File Name or Path (CWE ID 73)(58 flaws)

Description

This call contains a path manipulation flaw. The argument to the function is a filename constructed using untrusted input. If an attacker is allowed to specify all or part of the filename, it may be possible to gain unauthorized access to files on the server, including those outside the webroot, that would be normally be inaccessible to end users. The level of exposure depends on the effectiveness of input validation routines, if any.

Effort to Fix: 2 - Implementation error. Fix is approx. 6-50 lines of code. 1 day to fix.

Recommendations

Validate all untrusted input to ensure that it conforms to the expected format, using centralized data validation routines when possible. When using black lists, be sure that the sanitizing routine performs a sufficient number of iterations to remove all instances of disallowed characters.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2426	-	36	vlr.classlibrary.dll	bool FileExists(string) 13%	
2170	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	bool SplitChildDocumentsList_ITextSharp(string, System.Collections.Generic.List<IDX_GetChildPagesInformation_Result>, System.Collections.Generic.List<GetIndexingResultsForSplitting_Result>, string, ref System.Collections.Generic.List<System.Tuple<long, string, int>> /*1*/, ref System.Collections.Generic.List<System.Tuple<int, int, string, int, int, int>> /*1*/, FileOrder, int) 60%	
719	4	-	visionet.vlr.web.presentation.dll	.../communicationportalcontroller.cs 1625	
382	4	-	visionet.vlr.web.pres	.../communicationportalcontroller.cs 1645	

Flaw Id	Module #	Class #	Module	Location	Fix By
			entation.dll		
813	12	-	visionet.vlr.web.presentation.dll	.../models/filedownloadresult.cs 159	
819	12	-	visionet.vlr.web.presentation.dll	.../models/filedownloadresult.cs 234	
2306	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	int SplitFileITextSharp(ref string /*1*/, string, GetIndexingResultsForSplitting_Result, ref System.Collections.Generic.List<IDX_GetChildPagesInformation_Result> /*1*/, ref int) 56%	
572	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 4287	
2393	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 4295	
2387	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 4295	
743	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 4299	
2383	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 4300	
2381	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 4300	
636	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 4845	
887	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 6330	
552	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 7227	
2130	-	38	vlr.common.dll	string GenerateReport(ref System.Web.UI.Page /*1*/, CrystalDecisions.CrystalReports.Engine.ReportDocument, string) 53%	
2131	-	38	vlr.common.dll	string GenerateReport(ref System.Web.UI.Page /*1*/, CrystalDecisions.CrystalReports.Engine.ReportDocument, string) 73%	
2263	-	15	visionet.vlr.web.infrastructure.dll	string LoadDocumentFile(string, string, int, string, string, string) 67%	
2264	-	15	visionet.vlr.web.infrastructure.dll	string LoadDocumentFile(string, string, int, string, string, string) 71%	
2297	-	17	visionet.vlr.web.infrastructure.dll	Visionet.VLR.Model.Enums.EventActionStatus SaveLoanDataReview(System.Collections.Generic.List<Visionet.VLR.Model.TaskSaveTemplate>, string, string, string, int, int, bool, int, bool, bool, (3 more parameters)) 44%	
2298	-	17	visionet.vlr.web.infrastructure.dll	Visionet.VLR.Model.Enums.EventActionStatus SaveLoanDataReview(System.Collections.Generic.List<Visionet.VLR.Model.TaskSaveTemplate>, string, string, string, int, int, bool, int, bool, bool, (3 more parameters)) 53%	

Flaw Id	Module #	Class #	Module	Location	Fix By
2299	-	17	visionet.vlr.web.infrastructure.dll	Visionet.VLR.Model.Enums.EventActionStatus SaveLoanDataReview(System.Collections.Generic.List<Visionet.VLR.Model.TaskSaveTemplate>, string, string, string, int, int, bool, int, bool, bool, (3 more parameters)) 82%	
2300	-	17	visionet.vlr.web.infrastructure.dll	Visionet.VLR.Model.Enums.EventActionStatus SaveLoanDataReview(System.Collections.Generic.List<Visionet.VLR.Model.TaskSaveTemplate>, string, string, string, int, int, bool, int, bool, bool, (3 more parameters)) 90%	
2301	-	17	visionet.vlr.web.infrastructure.dll	Visionet.VLR.Model.Enums.EventActionStatus SaveLoanDataReview_V2(System.Collections.Generic.List<Visionet.VLR.Model.TaskSaveTemplate>, string, string, string, int, int, bool, int, bool, bool, (4 more parameters)) 44%	
2302	-	17	visionet.vlr.web.infrastructure.dll	Visionet.VLR.Model.Enums.EventActionStatus SaveLoanDataReview_V2(System.Collections.Generic.List<Visionet.VLR.Model.TaskSaveTemplate>, string, string, string, int, int, bool, int, bool, bool, (4 more parameters)) 53%	
2303	-	17	visionet.vlr.web.infrastructure.dll	Visionet.VLR.Model.Enums.EventActionStatus SaveLoanDataReview_V2(System.Collections.Generic.List<Visionet.VLR.Model.TaskSaveTemplate>, string, string, string, int, int, bool, int, bool, bool, (4 more parameters)) 82%	
2304	-	17	visionet.vlr.web.infrastructure.dll	Visionet.VLR.Model.Enums.EventActionStatus SaveLoanDataReview_V2(System.Collections.Generic.List<Visionet.VLR.Model.TaskSaveTemplate>, string, string, string, int, int, bool, int, bool, bool, (4 more parameters)) 90%	
2296	-	16	visionet.vlr.web.infrastructure.dll	void CombineMultiplePDFs(string[], string) 13%	
2288	-	16	visionet.vlr.web.infrastructure.dll	void CombineMultiplePDFs(System.Collections.Generic.List<string>, string) 10%	
2294	-	16	visionet.vlr.web.infrastructure.dll	void CombinePDFs(string, string, string) 39%	
2295	-	16	visionet.vlr.web.infrastructure.dll	void CombinePDFs(string, string, string) 59%	
2172	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	void DeleteFiles(System.Collections.Generic.List<System.Tuple<long, string, int>>) 30%	
2167	-	11	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	void DeleteMergingDocuments(string, System.Collections.Generic.List<DocumentsToMerge_Result>) 69%	
2175	-	13	visionet.vlr.web.presentation.dll/visionet.	void DeleteMergingDocuments(string, System.Collections.Generic.List<Get_Do	

Flaw Id	Module #	Class #	Module	Location	Fix By
			vlr.ocrfileorder.backend.dll	cumentsListForMerging_Result>) 67%	
2169	-	12	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	void DeleteMergingDocuments(string, System.Collections.Generic.List<SP_GetDocumentsToMergeForLenderLoanNumber_Result>) 69%	
2171	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	void DeleteParentIndexedFile(long, string, int) 12%	
2277	-	16	visionet.vlr.web.infrastructure.dll	void ExportToPdf(System.Data.DataSet, string) 0%	
2305	-	17	visionet.vlr.web.infrastructure.dll	void ExportToPdf(System.Data.DataTable, System.Data.DataTable, string, string) 1%	
2278	-	16	visionet.vlr.web.infrastructure.dll	void ExportToPdfFaxCoverLetters(string, string, string, string, string, string, string, string, string, string, (8 more parameters)) 33%	
2178	-	36	vlr.classlibrary.dll	void FTPFiles(int) 70%	
2180	-	36	vlr.classlibrary.dll	void FTPFiles(int) 74%	
2179	-	36	vlr.classlibrary.dll	void FTPFiles(int) 74%	
2286	-	16	visionet.vlr.web.infrastructure.dll	void GenerateFaxCoverLetters(string, string, string, string, string, string, string, string, string, string, (7 more parameters)) 72%	
2287	-	16	visionet.vlr.web.infrastructure.dll	void GenerateFaxCoverLetters(string, string, string, string, string, string, string, string, string, string, (7 more parameters)) 96%	
2166	-	11	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	void MergeDocument(System.Collections.Generic.List<DocumentsToMerge_Result>, IDX_AutoMergeDocuments_Result, string, int) 25%	
2174	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	void MergeDocument(System.Collections.Generic.List<Get_DocumentsListForMerging_Result>, string, int, string, int[], System.Nullable<int>) 30%	
2168	-	12	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	void MergeDocument(System.Collections.Generic.List<SP_GetDocumentsToMergeForLenderLoanNumber_Result>, IDX_AutoMergeDocumentsForLenderLoanNumber_Result, string, int) 28%	
2173	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	void RecreateParentIndexedFiles_ITextSharp(string, System.Collections.Generic.List<System.Tuple<int, int>>, ref System.Collections.Generic.List<System.Tuple<int, int, string, int, int, int>> /*1*/, int) 27%	
2289	-	16	visionet.vlr.web.infrastructure.dll	void RotatePDF(string, string, string, string) 36%	
2290	-	16	visionet.vlr.web.infra	void RotatePDF(string, string, string,	

Flaw Id	Module #	Class #	Module	Location	Fix By
			structure.dll	string) 78%	
2291	-	16	visionet.vlr.web.infrastructure.dll	void RotatePDF(string, string, string, string) 79%	
2292	-	16	visionet.vlr.web.infrastructure.dll	void RotatePDF(string, string, string, string) 79%	
2293	-	16	visionet.vlr.web.infrastructure.dll	void RotatePDF(string, string, string, string) 81%	
2177	-	35	vlr.classlibrary.dll	void UploadFile(string, string, bool) 54%	
2265	-	15	visionet.vlr.web.infrastructure.dll	void UploadOrder(int, string, string, string, string) 42%	
2182	-	36	vlr.classlibrary.dll	void WriteDataIntoFile(string, string) 29%	
2183	-	36	vlr.classlibrary.dll	void WriteDataIntoFile(string, string) 46%	

Information Leakage(1 flaw)

Description

An information leak is the intentional or unintentional disclosure of information that is either regarded as sensitive within the product's own functionality or provides information about the product or its environment that could be useful in an attack. Information leakage issues are commonly overlooked because they cannot be used to directly exploit the application. However, information leaks should be viewed as building blocks that an attacker uses to carry out other, more complicated attacks.

There are many different types of problems that involve information leaks, with severities that can range widely depending on the type of information leaked and the context of the information with respect to the application. Common sources of information leakage include, but are not limited to:

* Source code disclosure
* Browsable directories
* Log files or backup files in web-accessible directories
* Unfiltered backend error messages
* Exception stack traces
* Server version information
* Transmission of uninitialized memory containing sensitive data

Recommendations

Configure applications and servers to return generic error messages and to suppress stack traces from being displayed to end users. Ensure that errors generated by the application do not provide insight into specific backend issues.

Remove all backup files, binary archives, alternate versions of files, and test files from web-accessible directories of production servers. The only files that should be present in the application's web document root are files required by the application. Ensure that deployment procedures include the removal of these file types by an administrator. Keep web and application servers fully patched to minimize exposure to publicly-disclosed information leakage vulnerabilities.

Associated Flaws by CWE ID:

→ Server-Side Request Forgery (SSRF) (CWE ID 918)(1 flaw)

Description

The web server receives a URL or similar request from an upstream component and retrieves the contents of this URL, but it does not sufficiently ensure that the request is being sent to the expected destination.

Effort to Fix: 1 - Trivial implementation error. Fix is up to 5 lines of code. One hour or less to fix.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
632	17	-	visionet.vlr.web.presentation.dll	.../common/logincontroller.cs 617	

→ Insufficient Input Validation(2 flaws)

Description

Weaknesses in this category are related to an absent or incorrect protection mechanism that fails to properly validate input that can affect the control flow or data flow of a program.

Recommendations

Validate input from untrusted sources before it is used. The untrusted data sources may include HTTP requests, file systems, databases, and any external systems that provide data to the application. In the case of HTTP requests, validate all parts of the request, including headers, form fields, cookies, and URL components that are used to transfer information from the browser to the server side application.

Duplicate any client-side checks on the server side. This should be simple to implement in terms of time and difficulty, and will greatly reduce the likelihood of insecure parameter values being used in the application.

Associated Flaws by CWE ID:

→ URL Redirection to Untrusted Site ('Open Redirect') (CWE ID 601)(2 flaws)

Description

A web application accepts a untrusted input that specifies a link to an external site, and uses that link to generate a redirect. This enables phishing attacks.

Effort to Fix: 2 - Implementation error. Fix is approx. 6-50 lines of code. 1 day to fix.

Recommendations

Always validate untrusted input to ensure that it conforms to the expected format, using centralized data validation routines when possible. Check the supplied URL against a whitelist of approved URLs or domains before redirecting.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
378	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 3711	
2396	17	-	visionet.vlr.web.pres	.../common/logincontroller.cs 158	

Flaw Id	Module #	Class #	Module	Location	Fix By
			entation.dll		

Low (98 flaws)

Cryptographic Issues(6 flaws)

Description

Applications commonly use cryptography to implement authentication mechanisms and to ensure the confidentiality and integrity of sensitive data, both in transit and at rest. The proper and accurate implementation of cryptography is extremely critical to its efficacy. Configuration or coding mistakes as well as incorrect assumptions may negate a large degree of the protection it affords, leaving the crypto implementation vulnerable to attack.

Common cryptographic mistakes include, but are not limited to, selecting weak keys or weak cipher modes, unintentionally exposing sensitive cryptographic data, using predictable entropy sources, and mismanaging or hard-coding keys.

Developers often make the dangerous assumption that they can improve security by designing their own cryptographic algorithm; however, one of the basic tenets of cryptography is that any cipher whose effectiveness is reliant on the secrecy of the algorithm is fundamentally flawed.

Recommendations

Select the appropriate type of cryptography for the intended purpose. Avoid proprietary encryption algorithms as they typically rely on "security through obscurity" rather than sound mathematics. Select key sizes appropriate for the data being protected; for high assurance applications, 256-bit symmetric keys and 2048-bit asymmetric keys are sufficient. Follow best practices for key storage, and ensure that plaintext data and key material are not inadvertently exposed.

Associated Flaws by CWE ID:

Generation of Predictable IV with CBC Mode (CWE ID 329)(6 flaws)

Description

Not using a random initialization Vector (IV) with Cipher Block Chaining (CBC) Mode or other feedback-driven modes causes algorithms to be susceptible to dictionary attacks.

Effort to Fix: 2 - Implementation error. Fix is approx. 6-50 lines of code. 1 day to fix.

Recommendations

Ensure that IVs are unique and unpredictable. They do not need to be kept secret.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2389	18	-	visionet.vlr.web.presentation.dll	projects/.../passwordhash.cs 151	
2382	18	-	visionet.vlr.web.presentation.dll	projects/.../passwordhash.cs 192	
2333	-	10	visionet.vlr.common.dll	string Decrypt(string) 39%	
2152	-	9	visionet.vlr.web.presentation.dll/visionet.l	string Decrypt(string) 39%	

© 2021 Veracode, Inc.

Flaw Id	Module #	Class #	Module	Location	Fix By
			ogging.dll		
2332	-	10	visionet.vlr.common.dll	string Encrypt(string) 32%	
2151	-	9	visionet.vlr.web.presentation.dll/visionet.logging.dll	string Encrypt(string) 32%	

Information Leakage(34 flaws)

Description

An information leak is the intentional or unintentional disclosure of information that is either regarded as sensitive within the product's own functionality or provides information about the product or its environment that could be useful in an attack. Information leakage issues are commonly overlooked because they cannot be used to directly exploit the application. However, information leaks should be viewed as building blocks that an attacker uses to carry out other, more complicated attacks.

There are many different types of problems that involve information leaks, with severities that can range widely depending on the type of information leaked and the context of the information with respect to the application. Common sources of information leakage include, but are not limited to:

- Source code disclosure
- Browsable directories
- Log files or backup files in web-accessible directories
- Unfiltered backend error messages
- Exception stack traces
- Server version information
- Transmission of uninitialized memory containing sensitive data

Recommendations

Configure applications and servers to return generic error messages and to suppress stack traces from being displayed to end users. Ensure that errors generated by the application do not provide insight into specific backend issues.

Remove all backup files, binary archives, alternate versions of files, and test files from web-accessible directories of production servers. The only files that should be present in the application's web document root are files required by the application. Ensure that deployment procedures include the removal of these file types by an administrator. Keep web and application servers fully patched to minimize exposure to publicly-disclosed information leakage vulnerabilities.

Associated Flaws by CWE ID:

Generation of Error Message Containing Sensitive Information (CWE ID 209)(12 flaws)

Description

The software generates an error message that includes sensitive information about its environment, users, or associated data. The sensitive information may be valuable information on its own (such as a password), or it may be useful for launching other, more deadly attacks. If an attack fails, an attacker may use error information provided by the server to launch another more focused attack. For example, file locations disclosed by an exception stack trace may be leveraged by an attacker to exploit a path traversal issue elsewhere in the application.

Effort to Fix: 1 - Trivial implementation error. Fix is up to 5 lines of code. One hour or less to fix.

Recommendations

Ensure that only generic error messages are returned to the end user that do not reveal any additional details.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
389	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 3597	
794	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 3891	
591	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 3918	
597	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 3954	
650	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 6346	
870	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 7010	
706	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 7156	
729	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 7901	
369	10	-	visionet.vlr.web.presentation.dll	.../exceptionmanagementcontroller.cs 1320	
2097	-	8	visionet.vlr.web.presentation.dll/visionet.logging.dll	void SendMail(string, LogType) 84%	
2098	-	8	visionet.vlr.web.presentation.dll/visionet.logging.dll	void SendMail(string, LogType) 84%	
2196	-	37	vlr.common.dll	void ShowMessage(System.Web.UI.WebControls.Label, MessageType, System.Exception, string) 63%	

→ Insertion of Sensitive Information Into Sent Data (CWE ID 201)(22 flaws)

Description

Sensitive information may be exposed as a result of outbound network connections made by the application.

Effort to Fix: 2 - Implementation error. Fix is approx. 6-50 lines of code. 1 day to fix.

Recommendations

Ensure that the transfer of sensitive data is intended and that it does not violate application security policy or user expectations.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
691	1	-	visionet.vlr.web.pres	.../admin/admincontroller.cs 5438	

Flaw Id	Module #	Class #	Module	Location	Fix By
			entation.dll		
531	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 8230	
2187	-	35	vlr.classlibrary.dll	bool Login() 12%	
877	4	-	visionet.vlr.web.presentation.dll	.../communicationportalcontroller.cs 223	
838	6	-	visionet.vlr.web.presentation.dll	.../dashboardcontroller.cs 243	
829	6	-	visionet.vlr.web.presentation.dll	.../dashboardcontroller.cs 243	
587	15	-	visionet.vlr.web.presentation.dll	.../loanreviewcontroller.cs 2281	
2395	17	-	visionet.vlr.web.presentation.dll	.../common/logincontroller.cs 2198	
2390	17	-	visionet.vlr.web.presentation.dll	.../common/logincontroller.cs 2198	
2391	17	-	visionet.vlr.web.presentation.dll	.../common/logincontroller.cs 2238	
2392	17	-	visionet.vlr.web.presentation.dll	.../common/logincontroller.cs 2238	
2193	-	31	vlr.classlibrary.dll	void EmailLoanRequestReminder(VLR.Common.LoanRequestReminderInfo) 97%	
2194	-	31	vlr.classlibrary.dll	void EmailLoanRequestReminder(VLR.Common.LoanRequestReminderInfo) 97%	
2188	-	35	vlr.classlibrary.dll	void SendCommand(string) 73%	
2191	-	32	vlr.classlibrary.dll	void SendEmail(string, string, string, string, string, string, string, string, string, string[], (1 more parameter)) 98%	
2192	-	32	vlr.classlibrary.dll	void SendEmail(string, string, string, string, string, string, string, string, string, string[], (1 more parameter)) 98%	
2184	-	7	visionet.email.dll	void SendEmail(System.Net.Mail.MailMessage) 93%	
2185	-	7	visionet.email.dll	void SendEmail(System.Net.Mail.MailMessage) 93%	
2186	-	31	vlr.classlibrary.dll	void SendLoanRequestEmail(VLR.Common.LoanRequest) 97%	
2195	-	31	vlr.classlibrary.dll	void SendLoanRequestEmail(VLR.Common.LoanRequest) 97%	
2082	-	8	visionet.vlr.web.presentation.dll/visionet.logging.dll	void SendMail(string, LogType) 84%	
2083	-	8	visionet.vlr.web.presentation.dll/visionet.logging.dll	void SendMail(string, LogType) 84%	

→ Insufficient Input Validation(58 flaws)

Description

Weaknesses in this category are related to an absent or incorrect protection mechanism that fails to properly validate input that can affect the control flow or data flow of a program.

Recommendations

Validate input from untrusted sources before it is used. The untrusted data sources may include HTTP requests, file systems, databases, and any external systems that provide data to the application. In the case of HTTP requests, validate all parts of the request, including headers, form fields, cookies, and URL components that are used to transfer information from the browser to the server side application.

Duplicate any client-side checks on the server side. This should be simple to implement in terms of time and difficulty, and will greatly reduce the likelihood of insecure parameter values being used in the application.

Associated Flaws by CWE ID:

→ ASP.NET Misconfiguration: Improper Model Validation (CWE ID 1174)(58 flaws)

Description

The ASP.NET application does not use, or incorrectly uses, the model validation framework.

Effort to Fix: 3 - Complex implementation error. Fix is approx. 51-500 lines of code. Up to 5 days to fix.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2386	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 5050	
2394	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 8130	
2380	1	-	visionet.vlr.web.presentation.dll	.../admin/admincontroller.cs 10072	
439	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 20	
881	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 23	
652	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 24	
635	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 26	
775	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 27	
653	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 28	
508	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 29	
448	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 30	

Flaw Id	Module #	Class #	Module	Location	Fix By
660	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 33	
573	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 36	
367	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 40	
771	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 41	
797	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 52	
630	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 53	
763	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 54	
418	7	-	visionet.vlr.web.presentation.dll	.../models/dataimportmodel.cs 56	
826	9	-	visionet.vlr.web.presentation.dll	projects/.../models/entitymodel.cs 12	
639	9	-	visionet.vlr.web.presentation.dll	projects/.../models/entitymodel.cs 13	
818	9	-	visionet.vlr.web.presentation.dll	projects/.../models/entitymodel.cs 14	
861	9	-	visionet.vlr.web.presentation.dll	projects/.../models/entitymodel.cs 15	
534	9	-	visionet.vlr.web.presentation.dll	projects/.../models/entitymodel.cs 16	
593	9	-	visionet.vlr.web.presentation.dll	projects/.../models/entitymodel.cs 17	
836	9	-	visionet.vlr.web.presentation.dll	projects/.../models/entitymodel.cs 18	
2251	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_AtCloseOrderNo() 0%	
2254	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_BatchName() 0%	
2250	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_ClientFilePath() 0%	
2255	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_ClientLoanNumber() 0%	
2242	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_CustomerName() 0%	
2237	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_CustomerOrderID() 0%	

Flaw Id	Module #	Class #	Module	Location	Fix By
2238	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_FileName() 0%	
2239	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_FileNameAlias() 0%	
2244	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_LoanNumber() 0%	
2252	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_OPACode() 0%	
2240	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_PhysicalPath() 0%	
2245	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_ProcessingMachineName() 0%	
2246	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_ProductCode() 0%	
2248	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_SchemeName() 0%	
2247	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_SequenceID() 0%	
2241	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_SourceProject() 0%	
2249	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_SplittedStatus() 0%	
2253	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_TransactionId() 0%	
2243	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_TransmissionCode() 0%	
2236	-	14	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_VSIOrderID() 0%	
524	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 12	
451	23	-	visionet.vlr.web.pres	.../usermaintainprofilemodel.cs 13	

Flaw Id	Module #	Class #	Module	Location	Fix By
			entation.dll		
873	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 14	
842	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 15	
905	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 16	
867	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 22	
846	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 31	
897	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 32	
365	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 33	
911	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 34	
364	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 35	
774	23	-	visionet.vlr.web.presentation.dll	.../usermaintainprofilemodel.cs 36	

Very Low (0 flaws)

No flaws of this type were found

Info (2 flaws)

Code Quality(2 flaws)

Description

Code quality issues stem from failure to follow good coding practices and can lead to unpredictable behavior. These may include but are not limited to:

- Neglecting to remove debug code or dead code
- Improper resource management, such as using a pointer after it has been freed
- Using the incorrect operator to compare objects
- Failing to follow an API or framework specification
- Using a language feature or API in an unintended manner

While code quality flaws are generally less severe than other categories and usually are not directly exploitable, they may serve as indicators that developers are not following practices that increase the reliability and security of an application. For an attacker, code quality issues may provide an opportunity to stress the application in unexpected ways.

Recommendations

The wide variance of code quality issues makes it impractical to generalize how these issues should be addressed. Refer to individual categories for specific recommendations.

Associated Flaws by CWE ID:

→ Improper Resource Shutdown or Release (CWE ID 404)(2 flaws)

Description

The application fails to release (or incorrectly releases) a system resource before it is made available for re-use. This condition often occurs with resources such as database connections or file handles. Most unreleased resource issues result in general software reliability problems, but if an attacker can intentionally trigger a resource leak, it may be possible to launch a denial of service attack by depleting the resource pool.

Effort to Fix: 2 - Implementation error. Fix is approx. 6-50 lines of code. 1 day to fix.

Recommendations

When a resource is created or allocated, the developer is responsible for properly releasing the resource as well as accounting for all potential paths of expiration or invalidation. Ensure that all code paths properly release resources.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2189	-	34	vlr.classlibrary.dll	void !ctor() 70%	
2190	-	33	vlr.classlibrary.dll	void !ctor() 70%	

© 2021 Veracode, Inc.

Flaws in Common Modules

This section highlights the score impact of flaws in common modules in this application.

Module: Visionet.VLR.Web.Infrastructure.dll Used by 2 executables; score impact: 5

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
visionet_vlr_web_infrastructure_dll.Visionet.VLR.Web.Infrastructure.CommonServiceHelper					
CombinePDFs 59%	3	1	Directory Traversal	73	Neutral
CombinePDFs 39%	3	1	Directory Traversal	73	Neutral
CombineMultiplePDFs 13%	3	1	Directory Traversal	73	Neutral
GenerateFaxCoverLetters 72%	3	1	Directory Traversal	73	Neutral
RotatePDF 81%	3	1	Directory Traversal	73	Neutral
CombineMultiplePDFs 10%	3	1	Directory Traversal	73	Neutral
RotatePDF 78%	3	1	Directory Traversal	73	Neutral
RotatePDF 79%	3	2	Directory Traversal	73	Neutral
RotatePDF 36%	3	1	Directory Traversal	73	Neutral
GenerateFaxCoverLetters 96%	3	1	Directory Traversal	73	Neutral
ExportToPdfFaxCoverLetters 33%	3	1	Directory Traversal	73	Neutral
ExportToPdf 0%	3	1	Directory Traversal	73	Neutral
visionet_vlr_web_infrastructure_dll.Visionet.VLR.Web.Infrastructure.LoanReviewHelper					
SaveLoanDataReview_V2 53%	3	1	Directory Traversal	73	Neutral
SaveLoanDataReview 90%	3	1	Directory Traversal	73	Neutral
SaveLoanDataReview_V2 82%	3	1	Directory Traversal	73	Neutral
SaveLoanDataReview 44%	3	1	Directory Traversal	73	Neutral
SaveLoanDataReview_V2 90%	3	1	Directory Traversal	73	Neutral
SaveLoanDataReview 53%	3	1	Directory Traversal	73	Neutral
ExportToPdf 1%	3	1	Directory Traversal	73	Neutral
SaveLoanDataReview 82%	3	1	Directory Traversal	73	Neutral
SaveLoanDataReview_V2 44%	3	1	Directory Traversal	73	Neutral
visionet_vlr_web_infrastructure_dll.Visionet.VLR.Infrastructure.LoanDocumentsHelper					
LoadDocumentFile 71%	3	1	Directory Traversal	73	Neutral
UploadOrder 42%	3	1	Directory Traversal	73	Neutral
LoadDocumentFile 67%	3	1	Directory Traversal	73	Neutral

Module: Visionet.VLR.OCRFileOrder.Backend.dll Used by 4 executables; score impact: 4

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.BL.OCRFileOrderBL					
DeleteMergingDocuments 67%	3	1	Directory Traversal	73	Neutral
DeleteParentIndexedFile 12%	3	1	Directory Traversal	73	Neutral
RecreateParentIndexedFiles_ITextSharp 27%	3	1	Directory Traversal	73	Neutral
SplitChildDocumentsList_ITextSharp 60%	3	1	Directory Traversal	73	Neutral

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
DeleteFiles 30%	3	1	Directory Traversal	73	Neutral
SplitFileITextSharp 56%	3	1	Directory Traversal	73	Neutral
MergeDocument 30%	3	1	Directory Traversal	73	Neutral
visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.FileOrder					
get_SplittedStatus 0%	2	1	Insufficient Input Validation	1174	Neutral
get_PhysicalPath 0%	2	1	Insufficient Input Validation	1174	Neutral
get_FileName 0%	2	1	Insufficient Input Validation	1174	Neutral
get_SequenceID 0%	2	1	Insufficient Input Validation	1174	Neutral
get_ClientLoanNumber 0%	2	1	Insufficient Input Validation	1174	Neutral
get_ProductCode 0%	2	1	Insufficient Input Validation	1174	Neutral
get_TransactionId 0%	2	1	Insufficient Input Validation	1174	Neutral
get_LoanNumber 0%	2	1	Insufficient Input Validation	1174	Neutral
get_SourceProject 0%	2	1	Insufficient Input Validation	1174	Neutral
get_FileNameAlias 0%	2	1	Insufficient Input Validation	1174	Neutral
get_BatchName 0%	2	1	Insufficient Input Validation	1174	Neutral
get_ClientFilePath 0%	2	1	Insufficient Input Validation	1174	Neutral
get_SchemeName 0%	2	1	Insufficient Input Validation	1174	Neutral
get_CustomerOrderID 0%	2	1	Insufficient Input Validation	1174	Neutral
get_AtCloseOrderNo 0%	2	1	Insufficient Input Validation	1174	Neutral
get_OPACode 0%	2	1	Insufficient Input Validation	1174	Neutral
get_VSIOrderID 0%	2	1	Insufficient Input Validation	1174	Neutral
get_CustomerName 0%	2	1	Insufficient Input Validation	1174	Neutral
get_TransmissionCode 0%	2	1	Insufficient Input Validation	1174	Neutral
get_ProcessingMachineName 0%	2	1	Insufficient Input Validation	1174	Neutral
visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.AutoMergeDocumentBL					
MergeDocument 25%	3	1	Directory Traversal	73	Neutral
DeleteMergingDocuments 69%	3	1	Directory Traversal	73	Neutral
visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.BL.AutoMergeForLenderLoanNumberBL					
DeleteMergingDocuments 69%	3	1	Directory Traversal	73	Neutral
MergeDocument 28%	3	1	Directory Traversal	73	Neutral

Module: VLR.ClassLibrary.dll Used by 10 executables; score impact: 2

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
vlr_classlibrary_dll.cFTPService					
FileExists 13%	3	1	Directory Traversal	73	Neutral
WriteDataIntoFile 46%	3	1	Directory Traversal	73	Neutral
WriteDataIntoFile 29%	3	1	Directory Traversal	73	Neutral
FTPFiles 70%	3	1	Directory Traversal	73	Neutral
FTPFiles 74%	3	2	Directory Traversal	73	Neutral
vlr_classlibrary_dll.cFTP					

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
UploadFile 54%	3	1	Directory Traversal	73	Neutral
Login 12%	2	1	Information Leakage	201	Unlikely
SendCommand 73%	2	1	Information Leakage	201	Unlikely
vlr_classlibrary_dll.AsyncEmailManager					
SendLoanRequestEmail 97%	2	2	Information Leakage	201	Unlikely
EmailLoanRequestReminder 97%	2	2	Information Leakage	201	Unlikely
vlr_classlibrary_dll.VLR.ClassLibrary.EmailManager					
SendEmail 98%	2	2	Information Leakage	201	Unlikely
vlr_classlibrary_dll.VLR.TapeCracking.FileSourceDAL					
!ctor 70%	0	1	Code Quality	404	Neutral
vlr_classlibrary_dll.VLR.TapeCracking.cXLTemplateDAL					
!ctor 70%	0	1	Code Quality	404	Neutral

Module: VLR.Common.dll Used by 17 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
vlr_common_dll.VLR.Common.cReportManager					
GenerateReport 53%	3	1	Directory Traversal	73	Neutral
GenerateReport 73%	3	1	Directory Traversal	73	Neutral
vlr_common_dll.VLR.Common.CommonLib					
ShowMessage 63%	2	1	Information Leakage	209	Neutral

Module: Visionet.Logging.dll Used by 22 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
visionet_logging_dll.Visionet.Logging.SecurityManager					
Encrypt 32%	2	1	Cryptographic Issues	329	Neutral
Decrypt 39%	2	1	Cryptographic Issues	329	Neutral
visionet_logging_dll.Visionet.Logging.CustomLogging					
SendMail 84%	2	4	Information Leakage	209	Unlikely – Neutral

Module: Visionet.VLR.Common.dll Used by 7 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
visionet_vlr_common_dll.Visionet.VLR.Common.SecurityManager					
Encrypt 32%	2	1	Cryptographic Issues	329	Neutral
Decrypt 39%	2	1	Cryptographic Issues	329	Neutral

Module: Visionet.EMail.dll Used by 11 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
visionet_email_dll.Visionet.EMail.SMTPEmailer					

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
SendEmail 93%	2	2	Information Leakage	201	Unlikely

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: Changes from Last Scan

Latest Scan		Prior Scan	
Static Scan			
Scan Name:	7 Dec 2021 Static (2)	Scan Name:	7 Dec 2021 Static
Completed:	12/7/21	Completed:	12/7/21
Score:	76	Score:	69

Changes in scope of scan (static)

Removed modules

Module Name	Compiler	Operating Environment	Engine Version
Document.Splitting.Service.exe	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.API.Common.dll	MSIL_MSVC14_X86_64	Windows X86_64	20211110154823
Visionet.FTPOrdersController.Service.exe	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.NPFLoanFoldersGenerator.Service.exe	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.OCR.DataProcessing.Service.exe	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.OCR.Engine.Service.exe	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.OrderPlacement.Service.dll	MSIL_MSVC14_X86_64	Windows X86_64	20211110154823
Visionet.OutboundIntegration.Service.exe	MSIL_MSVC14_X86	Windows	20211110154823
Visionet.ThumbnailService.exe	MSIL_MSVC14_X86_64	Windows X86_64	20211110154823
VLRInboundService.exe	MSIL_MSVC14_X86	Windows	20211110154823

Flaws not detected in current scan

The following is a list of all flaws found in the prior scan of this application that were not detected in the current scan.

Medium (72 flaws)

→ Cross-Site Scripting (XSS)(4 flaws)

Associated Flaws by CWE ID:

→ Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) (CWE ID 80)(4 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2330	-	24	visionet.vlr.web.presentation.dll	string GetLoanDocumentImage() 86%	
2331	-	24	visionet.vlr.web.presentation.dll	string GetLoanDocumentImage() 98%	
2313	-	23	visionet.vlr.web.pres	string GetLoanDocumentImage(string,	

Flaw Id	Module #	Class #	Module	Location	Fix By
			entation.dll	string, string) 82%	
2314	-	23	visionet.vlr.web.presentation.dll	string GetLoanDocumentImage(string, string, string) 98%	

Directory Traversal(63 flaws)

Associated Flaws by CWE ID:

External Control of File Name or Path (CWE ID 73)(63 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2416	-	4	document.splitting.service.exe#1.0.0.0	bool MergeAndBookMarkPdfFiles(string[], string, string[], ref System.Collections.Generic.List<Common.FinalPackageDetails> /*1*/) 14%	
2417	-	4	document.splitting.service.exe#1.0.0.0	bool MergeAndNestedBookMarkPdfFiles(string[], string, string[], ref System.Collections.Generic.List<Common.FinalPackageDetails> /*1*/) 11%	
2156	-	4	document.splitting.service.exe#1.0.0.0	bool MergeAndNestedBookMarkPdfFiles(System.Collections.Generic.List<BookMark>, string, ref System.Collections.Generic.List<Common.FinalPackageDetails> /*1*/) 11%	
2415	-	2	document.splitting.service.exe#1.0.0.0	bool MergeFiles(string, System.Collections.Generic.List<string>) 12%	
2418	-	4	document.splitting.service.exe#1.0.0.0	bool MergeFiles(string, System.Collections.Generic.List<string>) 13%	
2419	-	3	document.splitting.service.exe#1.0.0.0	bool SingleIndexingResultOrderSplit(GetIndexingResults_forSplitting_Result, string, string, int) 24%	
2420	-	3	document.splitting.service.exe#1.0.0.0	bool SingleIndexingResultOrderSplit(GetIndexingResults_forSplitting_Result, string, string, int) 24%	
2327	-	23	visionet.vlr.web.presentation.dll	bool SplitListOfDocument(string, System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.GetIndexingResultsForSplitting_Result>, string, Visionet.VLR.OCRFileOrder.Backend.FileOrder, ref System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.IDX_GetChildPagesInformation_Result> /*1*/) 10%	
1132	2	-	visionet.outboundintegration.service.exe#1.0.0.0	projects/.../common/common.cs 59	

© 2021 Veracode, Inc.

Flaw Id	Module #	Class #	Module	Location	Fix By
1135	2	-	visionet.outboundintegration.service.exe#1.0.0.0	projects/.../common/common.cs 61	
1136	2	-	visionet.outboundintegration.service.exe#1.0.0.0	projects/.../common/common.cs 101	
110	8	-	visionet.api.common.dll	.../documentservicehelper.cs 346	
299	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 64	
1613	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 368	
304	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 372	
1610	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 514	
1609	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 580	
1612	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 657	
1611	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 757	
1608	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 828	
279	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 1145	
289	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 1157	
303	13	-	visionet.npfloanfoldersgenerator.servic	.../ftplogcontroller.cs 402	

© 2021 Veracode, Inc.

Flaw Id	Module #	Class #	Module	Location	Fix By
			e.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll		
283	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 404	
288	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 590	
296	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 590	
282	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 624	
285	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 860	
306	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 876	
297	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 876	
290	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 878	
300	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 884	
295	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 884	
294	13	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 913	
112	14	-	visionet.api.common.dll	.../ghostscriptsharp.cs 56	

Flaw Id	Module #	Class #	Module	Location	Fix By
165	19	-	visionet.orderplacement.service.dll	projects/.../common/pdfutil.cs 36	
1139	20	-	visionet.outboundintegration.service.exe #1.0.0.0	.../responsexmlpackage.cs 1433	
1131	21	-	visionet.outboundintegration.service.exe #1.0.0.0	.../searchsummarysheet.cs 27	
1138	21	-	visionet.outboundintegration.service.exe #1.0.0.0	.../searchsummarysheet.cs 139	
1134	21	-	visionet.outboundintegration.service.exe #1.0.0.0	.../searchsummarysheet.cs 149	
2364	-	1	visionet.orderplacement.service.dll/bytescoutocrextraction.dll	string ExtractDataFromPDF(object, string, string, string, string, int, bool) 26%	
2316	-	23	visionet.vlr.web.presentation.dll	string RenameDocument(string, System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.GetIndexingResultsForSplitting_Result>, string, Visionet.VLR.OCRFileOrder.Backend.FileOrder, ref System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.IDX_GetChildPagesInformation_Result> /*1*/, string, string, long, long) 34%	
2320	-	23	visionet.vlr.web.presentation.dll	string RenameDocument(string, System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.GetIndexingResultsForSplitting_Result>, string, Visionet.VLR.OCRFileOrder.Backend.FileOrder, ref System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.IDX_GetChildPagesInformation_Result> /*1*/, string, string, long, long) 70%	
2411	-	23	visionet.vlr.web.presentation.dll	string RenameDocument(string, System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.GetIndexingResultsForSplitting_Result>, string, Visionet.VLR.OCRFileOrder.Backend.FileOrder, ref System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.IDX_GetChildPagesInformation_Result> /*1*/, string, string, long, long) 70%	
2412	-	23	visionet.vlr.web.presentation.dll	string RenameDocument(string, System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.GetIndexingResultsForSplitting_Result>, string, Visionet.VLR.OCRFileOrder.Backend.FileOrder, ref System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.IDX_GetChildPagesInformation_Result> /*1*/, string, string, long, long) 74%	
2413	-	23	visionet.vlr.web.presentation.dll	string RenameDocument(string, System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.GetIndexingResultsForSplitting_Result>, string, Visionet.VLR.OCRFileOrder.Backend.Fil	

© 2021 Veracode, Inc.

Flaw Id	Module #	Class #	Module	Location	Fix By
				eOrder, ref System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.IDX_GetChildPagesInformation_Result> /*1*/, string, string, long, long) 81%	
2414	-	23	visionet.vlr.web.presentation.dll	string RenameDocument(string, System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.GetIndexingResultsForSplitting_Result>, string, Visionet.VLR.OCRFileOrder.Backend.FileOrder, ref System.Collections.Generic.List<Visionet.VLR.OCRFileOrder.Backend.IDX_GetChildPagesInformation_Result> /*1*/, string, string, long, long) 81%	
2164	-	3	document.splitting.service.exe#1.0.0.0	System.IO.FileStream GetDestinationFileStream(string, bool) 97%	
2161	-	3	document.splitting.service.exe#1.0.0.0	System.IO.MemoryStream GetSourceFileStream(string) 31%	
2329	-	23	visionet.vlr.web.presentation.dll	System.Web.Mvc.ActionResult DownloadFileBytesFromVLRDataPath() 71%	
2328	-	23	visionet.vlr.web.presentation.dll	System.Web.Mvc.ActionResult generateCheckListReportForTask(string, string, string, string) 91%	
2310	-	20	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult DeleteDocument(int, string, int, int) 33%	
2311	-	20	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult DeleteDocument(int, string, int, int) 76%	
2165	-	3	document.splitting.service.exe#1.0.0.0	void DeleteFiles(System.Collections.Generic.List<string>) 25%	
2363	-	1	visionet.orderplacement.service.dll/bytescoutocrextraction.dll	void GenerateTextFile(object, ref string /*1*/) 77%	
2163	-	3	document.splitting.service.exe#1.0.0.0	void GetSourceFileStream(ref System.IO.MemoryStream /*1*/, string) 27%	
2158	-	3	document.splitting.service.exe#1.0.0.0	void PerformSplittingV2(ProcessOrderSplitting_Result) 32%	
2275	-	18	visionet.vlr.web.presentation.dll	void SendFileInChunks(ref System.Web.Mvc.ControllerContext /*1*/) 10%	
2276	-	18	visionet.vlr.web.presentation.dll	void SendGZipCompressedFileInChunks(ref System.Web.Mvc.ControllerContext /*1*/) 61%	
149	24	-	visionet.orderplacement.service.dll	projects/.../common/xmlutil.cs 312	
153	24	-	visionet.orderplacement.service.dll	projects/.../common/xmlutil.cs 457	
154	24	-	visionet.orderplacement.service.dll	projects/.../common/xmlutil.cs 1101	
162	24	-	visionet.orderplace	projects/.../common/xmlutil.cs 1124	

Flaw Id	Module #	Class #	Module	Location	Fix By
			ment.service.dll		

Information Leakage(2 flaws)

Associated Flaws by CWE ID:

Server-Side Request Forgery (SSRF) (CWE ID 918)(2 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
158	3	-	visionet.orderplacement.service.dll	projects/.../common/common.cs 109	
2308	-	25	visionet.vlr.web.presentation.dll	Models.CaptchaResponse ValidateCaptcha(string) 99%	

Insufficient Input Validation(2 flaws)

Associated Flaws by CWE ID:

URL Redirection to Untrusted Site ('Open Redirect') (CWE ID 601)(2 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2307	-	25	visionet.vlr.web.presentation.dll	System.Web.Mvc.ActionResult RedirectToAction() 99%	
2315	-	23	visionet.vlr.web.presentation.dll	void ViewLoanDocumentImage() 92%	

Time and State(1 flaw)

Associated Flaws by CWE ID:

Insecure Temporary File (CWE ID 377)(1 flaw)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1142	-	6	visionet.thumbnailservice.exe#1.0.0.0/visionet.api.common.dll/kofax.omnipagecsdk.objects.dll	int TestRule(string) 5%	

Low (60 flaws)

Cryptographic Issues(3 flaws)

Associated Flaws by CWE ID:

Generation of Predictable IV with CBC Mode (CWE ID 329)(3 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2223	-	29	visionet.vlr.web.presentation.dll	string Decrypt(string) 39%	
2222	-	29	visionet.vlr.web.presentation.dll	string Encrypt(string) 32%	
1137	22	-	visionet.outboundintegration.service.exe#1.0.0.0/vsiencompassconnect.dll	.../symmetriccryptographer.cs 153	

Information Leakage(19 flaws)

Associated Flaws by CWE ID:

Generation of Error Message Containing Sensitive Information (CWE ID 209)(9 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2267	-	19	visionet.vlr.web.presentation.dll	string DDL_SUBTeamTaskAssociation(string) 98%	
2257	-	19	visionet.vlr.web.presentation.dll	string GetAttachedRulesByEvent(string, string, string, string, bool) 99%	
2269	-	19	visionet.vlr.web.presentation.dll	string GetCalendarConfiguration() 98%	
2259	-	19	visionet.vlr.web.presentation.dll	string GetClientRuleText(string) 97%	
2282	-	22	visionet.vlr.web.presentation.dll	string GetExceptionFieldMapping(Visionet.VLR.Model.DataFilter) 99%	
2262	-	19	visionet.vlr.web.presentation.dll	string GetReassignUsers(string, string, string, string, string, string, string) 98%	
2258	-	19	visionet.vlr.web.presentation.dll	string GetRuleText(string) 97%	
2256	-	19	visionet.vlr.web.presentation.dll	string GetTaskFieldMapping(Visionet.VLR.Model.DataFilter) 99%	
2268	-	19	visionet.vlr.web.presentation.dll	string OnChangeCustomObjectsDDL(string, string) 98%	

→ Insertion of Sensitive Information Into Sent Data (CWE ID 201)(10 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2261	-	19	visionet.vlr.web.presentation.dll	string DataTableToJSONWithStringBuilder(System.Data.DataTable) 99%	
2279	-	25	visionet.vlr.web.presentation.dll	string GetTokenInfo() 72%	
2280	-	25	visionet.vlr.web.presentation.dll	string GetTokenInfo() 72%	
2283	-	20	visionet.vlr.web.presentation.dll	string GetUploadPath() 98%	
2281	-	23	visionet.vlr.web.presentation.dll	string GetUploadPath() 98%	
2271	-	19	visionet.vlr.web.presentation.dll	string InsertRegions(int, System.Collections.Generic.List<Models.Regions>) 99%	
2285	-	21	visionet.vlr.web.presentation.dll	System.Collections.Generic.List<FilterField> GetUserSavedFilters(int) 74%	
2284	-	21	visionet.vlr.web.presentation.dll	System.Collections.Generic.List<FilterField> GetUserSavedFilters(int) 74%	
2274	-	26	visionet.vlr.web.presentation.dll	void MoveNext() 48%	
2273	-	26	visionet.vlr.web.presentation.dll	void MoveNext() 48%	

→ Insufficient Input Validation(38 flaws)

Associated Flaws by CWE ID:

→ ASP.NET Misconfiguration: Improper Model Validation (CWE ID 1174)(38 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
2221	-	28	visionet.vlr.web.presentation.dll	string get_AllowDelete() 0%	
2220	-	28	visionet.vlr.web.presentation.dll	string get_AlterScreenURL() 0%	
2232	-	30	visionet.vlr.web.presentation.dll	string get_AssignmentInformation() 0%	
2346	-	27	visionet.vlr.web.presentation.dll	string get_Code_Translation_Tepmlate_Name() 0%	
2348	-	27	visionet.vlr.web.presentation.dll	string get_Column_Alias() 0%	
2231	-	30	visionet.vlr.web.pres	string get_CreateAssignments() 0%	

Flaw Id	Module #	Class #	Module	Location	Fix By
			entation.dll		
2235	-	30	visionet.vlr.web.presentation.dll	string get_CreateOutsourceListing() 0%	
2230	-	30	visionet.vlr.web.presentation.dll	string get_CreateTransmittal() 0%	
2349	-	27	visionet.vlr.web.presentation.dll	string get_CSVSeperator() 0%	
2227	-	30	visionet.vlr.web.presentation.dll	string get_EmailAddress() 0%	
2351	-	28	visionet.vlr.web.presentation.dll	string get_Entity_Name() 0%	
2350	-	28	visionet.vlr.web.presentation.dll	string get_EntityId() 0%	
2234	-	30	visionet.vlr.web.presentation.dll	string get_EscalatedCalls() 0%	
2341	-	27	visionet.vlr.web.presentation.dll	string get_FileFieldName() 0%	
2218	-	28	visionet.vlr.web.presentation.dll	string get_FilterColumn_Name() 0%	
2219	-	28	visionet.vlr.web.presentation.dll	string get_FilterCoumnValue() 0%	
2225	-	30	visionet.vlr.web.presentation.dll	string get_FirstName() 0%	
2226	-	30	visionet.vlr.web.presentation.dll	string get_LastName() 0%	
2229	-	30	visionet.vlr.web.presentation.dll	string get_LoginName() 0%	
2228	-	30	visionet.vlr.web.presentation.dll	string get_Privilege() 0%	
2233	-	30	visionet.vlr.web.presentation.dll	string get_ReportAccess() 0%	
2347	-	27	visionet.vlr.web.presentation.dll	string get_Sheet_Name() 0%	
2342	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Datatype() 0%	
2343	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Default_Value() 0%	
2340	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Name() 0%	
2344	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Precesion() 0%	
2345	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Scale() 0%	
2337	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Name() 0%	
2339	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Table_Description() 0%	

Flaw Id	Module #	Class #	Module	Location	Fix By
2338	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Table_Name() 0%	
2352	-	28	visionet.vlr.web.presentation.dll	string get_Table_Name() 0%	
2336	-	27	visionet.vlr.web.presentation.dll	string get_TemplateDesc() 0%	
2335	-	27	visionet.vlr.web.presentation.dll	string get_TemplateName() 0%	
2334	-	27	visionet.vlr.web.presentation.dll	string get_TemplateTypeName() 0%	
2224	-	30	visionet.vlr.web.presentation.dll	string get_UserName() 0%	
2272	-	19	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult DataImportTemplateSave(Models.DataImportModel) 0%	
2270	-	19	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult MaintainUserProfileForUpdate(Models.UserMaintainProfileModel) 0%	
2260	-	19	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult SaveEntity(Models.EntityModel) 0%	

Info (5 flaws)

→ Code Quality(5 flaws)

Associated Flaws by CWE ID:

→ Improper Resource Shutdown or Release (CWE ID 404)(5 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1130	5	-	visionet.outboundintegration.service.exe#1.0.0.0/vsiencompassconnect.dll	temp/.../dal/dalbase.cs 46	
287	11	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 842	
1	-	5	visionet.thumbnailservice.exe#1.0.0.0/visionet.api.common.dll/kofax.omnipagecsdk.argtypes.dll	int ROpenStreamCallBack(string, int, ref System.UIntPtr) 53%	
2153	-	3	document.splitting.service.exe#1.0.0.0	void GetSourceFileStream(ref System.IO.MemoryStream /*1*/, string) 9%	
164	24	-	visionet.orderplacement.service.dll	projects/.../common/xmlutil.cs 59	

Appendix B: Referenced Source Files

Id	Filename	Path
1	admincontroller.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/controllers/admin/
2	common.cs	projects/pnq/source code dev/windows services/visionet.outboundintegration.service/visionet.outboundintegration.service/common/
3	common.cs	projects/pnq/source code dev/windows services/visionet.orderplacement.service/visionet.orderplacement.service/common/
4	communicationportalcontroller.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/controllers/communicationportal/
5	dalbase.cs	temp/pnc indexing/visionet.outboundintegration.service/vsiencompassconnect/dal/
6	dashboardcontroller.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/controllers/dashboard/
7	dataimportmodel.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/models/
8	documentservicehelper.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/
9	entitymodel.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/models/
10	exceptionmanagementcontroller.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/controllers/expmanagement/
11	fileconverter.cs	systems working/pnc indexing/source code dev/windows services/visionet.npfloanfoldersgenerator.service/visionet.fileconverter.lib/
12	filedownloadresult.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/models/
13	ftplogcontroller.cs	systems working/pnc indexing/source code dev/windows services/visionet.npfloanfoldersgenerator.service/visionet.npfloanfoldersgenerator.lib/
14	ghostscriptsharp.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/ghostscript/
15	loanreviewcontroller.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/controllers/loanreview/
16	loantaskcontroller.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/controllers/
17	logincontroller.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/controllers/common/
18	passwordhash.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/models/
19	pdfutil.cs	projects/pnq/source code dev/windows services/visionet.orderplacement.service/visionet.orderplacement.service/common/
20	responsexmlpackage.cs	projects/pnq/source code dev/windows services/visionet.outboundintegration.service/visionet.outboundintegration.service/businessclasses/
21	searchsummarysheet.cs	projects/pnq/source code dev/windows services/visionet.outboundintegration.service/visionet.outboundintegration.service/businessclasses/
22	symmetriccryptographer.cs	temp/pnc indexing/visionet.outboundintegration.service/vsiencompassconnect/dal/
23	usermaintainprofilemodel.cs	projects/pnq/source code dev/visionet.vlr/visionet.vlr.web.presentation/models/
24	xmlutil.cs	projects/pnq/source code dev/windows services/visionet.orderplacement.service/visionet.orderplacement.service/common/

Appendix C: Referenced Classpaths

Id	Path
1	bytescoutocrextraction_dll.BytesCoutOCRExtraction.PDFExtractor
2	document_splitting_service_exe.Document.Splitting.Service.DALHelper
3	document_splitting_service_exe.Document.Splitting.Service.Modules.SplitDocuments
4	document_splitting_service_exe.Document.Splitting.Service.Utility
5	kofax_omnipagecsdk_argtypes_dll.Kofax.OmniPageCSDK.ArgTypes.IOStreamCB
6	kofax_omnipagecsdk_objects_dll.Kofax.OmniPageCSDK.Objects.DataRule
7	visionet_email_dll.Visionet.EMail.SMTPEmailer
8	visionet_logging_dll.Visionet.Logging.CustomLogging
9	visionet_logging_dll.Visionet.Logging.SecurityManager
10	visionet_vlr_common_dll.Visionet.VLR.Common.SecurityManager
11	visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.AutoMergeDocumentBL
12	visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.BL.AutoMergeForLenderLoanNumberBL
13	visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.BL.OCRFileOrderBL
14	visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.FileOrder
15	visionet_vlr_web_infrastructure_dll.Visionet.VLR.Infrastructure.LoanDocumentsHelper
16	visionet_vlr_web_infrastructure_dll.Visionet.VLR.Web.Infrastructure.CommonServiceHelper
17	visionet_vlr_web_infrastructure_dll.Visionet.VLR.Web.Infrastructure.LoanReviewHelper
18	visionet_vlr_web_presentation_dll.FileDownloadInMvc3.Models.FileDownloadResult
19	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.AdminController
20	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.CommunicationPortalController
21	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.Dashboard.DashboardController
22	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.ExceptionManagementController
23	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.LoanReviewController
24	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.LoanTaskController
25	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.LoginController
26	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.LoginController._3CGetRefreshTokenInfo_3Ed__44
27	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Models.DataImportModel
28	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Models.EntityModel
29	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Models.PasswordHash
30	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Models.UserMaintainProfileModel
31	vlr_classlibrary_dll.AsyncEmailManager
32	vlr_classlibrary_dll.VLR.ClassLibrary.EmailManager
33	vlr_classlibrary_dll.VLR.TapeCracking.FileSourceDAL
34	vlr_classlibrary_dll.VLR.TapeCracking.cXLTemplateDAL
35	vlr_classlibrary_dll.cFTP
36	vlr_classlibrary_dll.cFTPService
37	vlr_common_dll.VLR.Common.CommonLib

Id	Path
38	vlr_common_dll.VLR.Common.cReportManager

© 2021 Veracode, Inc.

Appendix D: Dynamic Flaw Inventory

Rescan Status	Number of Flaws
All	0
New	0
Open and Reopened	0
Cannot Reproduce	0
Fixed	0

© 2021 Veracode, Inc.