Veracode Detailed Report

Application Security Report
As of 1 Sep 2021

Prepared for:	Visionet
Prepared on:	September 1, 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:	11 of 201 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.

Veracode Detailed Report
Application Security Report
As of 1 Sep 2021

Veracode Level: VL1
Rated: Sep 1, 2021

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

Scans Included in Report

Static Scan	Dynamic Scan	Manual Penetration Test
1 Sep 2021 Static Score: 57 Completed: 9/1/21	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)

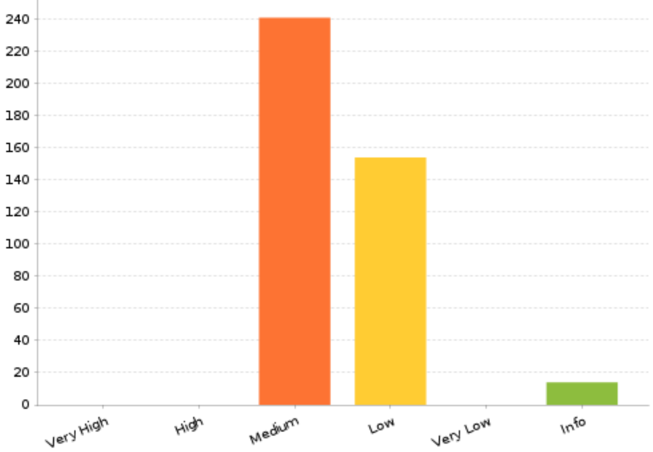
Impacts:Operational Risk (High), Financial Loss (High)

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 12/1/21. Please submit your application for Static Scan by the deadline and remediate the required detected flaws to conform to your assigned policy.

Flaws To Fix For Minimum Score

→ A rule in your policy requires a minimum score of 70. Fix 144 Medium flaws to reach a score of 70.

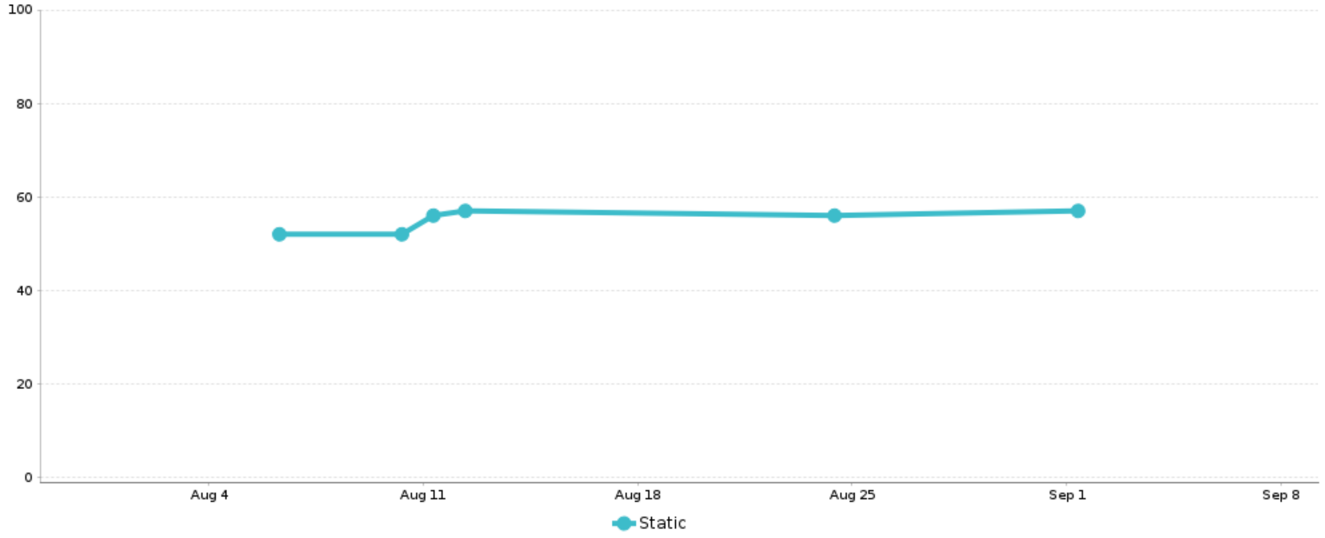
Flaw Severities

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

Longer Timeframe (6 - 12 months)

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

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: 20210819124611

The following modules were included in the application scan:

Module Name	Compiler	Operating Environment	Engine Version
Document.Splitting.Service.exe	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.API.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.FTPOrdersController.Service.exe	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.NPFLoanFoldersGenerator.Service.exe	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.OCR.DataProcessing.Service.exe	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.OCR.Engine.Service.exe	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.OrderPlacement.Service.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.OutboundIntegration.Service.exe	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.ThumbnailService.exe	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.VLR.Web.Presentation.dll	MSIL_MSVC14_X86	Windows	20210819124611
VLRInboundService.exe	MSIL_MSVC14_X86	Windows	20210819124611

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
Aspose.BarCode.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.Cells.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.Cells.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.PDF.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.PDF.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.PDF.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.PDF.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.PDF.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.PDF.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.Words.dll	MSIL_MSVC8_X86	Windows	20210819124611
Aspose.Words.dll	MSIL_MSVC6	Windows	20210819124611
Aspose.Words.dll	MSIL_MSVC8_X86	Windows	20210819124611
BGC128.dll	MSIL_MSVC6	Windows	20210819124611
BGC128.dll	MSIL_MSVC6	Windows	20210819124611
BouncyCastle.Crypto.dll	MSIL_MSVC14_X86	Windows	20210819124611
Bytescout.PDFExtractor.dll	MSIL_MSVC11_X86	Windows	20210819124611
Bytescout.PDFExtractor.OCRExtension.dll	MSIL_MSVC11_X86	Windows	20210819124611
BytesCoutOCRExtraction.DataAccess.dll	MSIL_MSVC14_X86	Windows	20210819124611
BytesCoutOCRExtraction.dll	MSIL_MSVC14_X86	Windows	20210819124611
Document.Splitting.Service.DAL.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
DotCMIS.dll	MSIL_MSVC8_X86	Windows	20210819124611
DotNetZip.dll	MSIL_MSVC14_X86	Windows	20210819124611
EPPlus.dll	MSIL_MSVC14_X86	Windows	20210819124611
EPPlus.dll	MSIL_MSVC14_X86	Windows	20210819124611
FluentFTP.dll	MSIL_MSVC14_X86	Windows	20210819124611

Module Name	Compiler	Operating Environment	Engine Version
Interop.Microsoft.Office.Core.dll	MSIL_MSVC6	Windows	20210819124611
Interop.Microsoft.Office.Core.dll	MSIL_MSVC6	Windows	20210819124611
itext.barcodes.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.barcodes.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.forms.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.forms.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.io.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.io.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.io.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.io.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.io.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.kernel.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.kernel.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.kernel.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.kernel.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.kernel.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.layout.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.layout.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.pdfa.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.pdfa.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.sign.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.sign.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.styledxmlparser.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.styledxmlparser.dll	MSIL_MSVC14_X86	Windows	20210819124611
itext.svg.dll	MSIL_MSVC14_X86	Windows	20210819124611

Module Name	Compiler	Operating Environment	Engine Version
itext.svg.dll	MSIL_MSVC14_X86	Windows	20210819124611
itextsharp.dll	MSIL_MSVC11_X86	Windows	20210819124611
itextsharp.dll	MSIL_MSVC8_X86	Windows	20210819124611
itextsharp.dll	MSIL_MSVC11_X86	Windows	20210819124611
itextsharp.dll	MSIL_MSVC11_X86	Windows	20210819124611
itextsharp.dll	MSIL_MSVC11_X86	Windows	20210819124611
itextsharp.dll	MSIL_MSVC11_X86	Windows	20210819124611
JS files within Packages.zip	JAVASCRIPT_5_1	JavaScript	20210819124611
Kofax.OmniPageCSDK.ArgTypes.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.ArgTypes.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.ArgTypes.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.ArgTypes.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.CAPI.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.CAPI.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.CAPI.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.CAPI.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.Objects.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.Objects.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.Objects.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Kofax.OmniPageCSDK.Objects.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
log4net.dll	MSIL_MSVC6	Windows	20210819124611
log4net.dll	MSIL_MSVC6	Windows	20210819124611
log4net.dll	MSIL_MSVC6	Windows	20210819124611
log4net.dll	MSIL_MSVC6	Windows	20210819124611
log4net.dll	MSIL_MSVC6	Windows	20210819124611

Module Name	Compiler	Operating Environment	Engine Version
log4net.dll	MSIL_MSVC6	Windows	2021081912 4611
log4net.dll	MSIL_MSVC6	Windows	2021081912 4611
log4net.dll	MSIL_MSVC6	Windows	2021081912 4611
log4net.dll	MSIL_MSVC6	Windows	2021081912 4611
log4net.dll	MSIL_MSVC6	Windows	2021081912 4611
Microsoft.Owin.Cors.dll	MSIL_MSVC14_X86	Windows	2021081912 4611
MismoProject.dll	MSIL_MSVC14_X86	Windows	2021081912 4611
MyGeneration.dOOdads.dll	MSIL_MSVC14_X86	Windows	2021081912 4611
Nest.dll	MSIL_MSVC11_X86	Windows	2021081912 4611
NHunspell.dll	MSIL_CPP_MSVC9_X86_64	Windows X86_64	2021081912 4611
NReco.LambdaParser.dll	MSIL_MSVC14_X86	Windows	2021081912 4611
Nuance.OmniPage.CSDK.ArgTypes.dll	MSIL_MSVC14_X86_64	Windows X86_64	2021081912 4611
Nuance.OmniPage.CSDK.ArgTypes.dll	MSIL_MSVC11_X86_64	Windows X86_64	2021081912 4611
Nuance.OmniPage.CSDK.ArgTypes.dll	MSIL_MSVC14_X86_64	Windows X86_64	2021081912 4611
Nuance.OmniPage.CSDK.CAPI.dll	MSIL_MSVC11_X86_64	Windows X86_64	2021081912 4611
Nuance.OmniPage.CSDK.CAPI.dll	MSIL_MSVC14_X86_64	Windows X86_64	2021081912 4611
Nuance.OmniPage.CSDK.CAPI.dll	MSIL_MSVC14_X86_64	Windows X86_64	2021081912 4611
Nuance.OmniPage.CSDK.Objects.dll	MSIL_MSVC11_X86_64	Windows X86_64	2021081912 4611
Nuance.OmniPage.CSDK.Objects.dll	MSIL_MSVC14_X86_64	Windows X86_64	2021081912 4611
Nuance.OmniPage.CSDK.Objects.dll	MSIL_MSVC14_X86_64	Windows X86_64	2021081912 4611
PdfSharp-WPF.dll	MSIL_MSVC8_X86	Windows	2021081912 4611
PdfSharp-WPF.dll	MSIL_MSVC11_X86	Windows	2021081912 4611
PdfSharp.Charting.dll	MSIL_MSVC14_X86	Windows	2021081912 4611
PdfSharp.dll	MSIL_MSVC14_X86	Windows	2021081912 4611
PdfSharp.dll	MSIL_MSVC14_X86	Windows	2021081912 4611

Module Name	Compiler	Operating Environment	Engine Version
PdfSharp.dll	MSIL_MSVC14_X86	Windows	20210819124611
PdfSharp.Xps.dll	MSIL_MSVC11_X86	Windows	20210819124611
RecPDFNet.dll	MSIL_MSVC11_X86_64	Windows X86_64	20210819124611
RestClient.Net.dll	MSIL_MSVC14_X86	Windows	20210819124611
System.Web.Mvc.dll	MSIL_MSVC14_X86	Windows	20210819124611
TallComponents.PDF.Kit.dll	MSIL_MSVC8_X86	Windows	20210819124611
TwoFactorAuth.Net.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.API.Common.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.API.Common.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.Bre.DAL.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Bre.DAL.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Bre.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Bre.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Bre.Interface.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Bre.Interface.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Correspondence.Common.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Correspondence.Common.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Correspondence.DAL.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Correspondence.DAL.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Correspondence.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Correspondence.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.EMail.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.EMail.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Facade.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Facade.dll	MSIL_MSVC14_X86	Windows	20210819124611

Module Name	Compiler	Operating Environment	Engine Version
Visionet.Facade.XmlSerializers.dll	MSIL_MSVC11_X86	Windows	20210819124611
Visionet.FileConverter.Lib.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.FTPOrdersController.DAL.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.FTPOrdersController.Lib.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.LenderQB.OCRVendor.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.LenderQB.OCRVendor.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Logging.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.NPFLoanFoldersGenerator.DAL.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.NPFLoanFoldersGenerator.Lib.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.OCR.Common.dll	MSIL_MSVC11_X86	Windows	20210819124611
Visionet.OCR.DataProcessing.Business.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.OCR.DataProcessing.DataAccess.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.OCR.Engine.Core.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.OCR.Engine.DataAccess.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.OCR.Engine.DataStructures.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611

Module Name	Compiler	Operating Environment	Engine Version
Visionet.OrderPlacement.Service.Library.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Security.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Security.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.SmartOCR.Engine.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Visiflow.BLL.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Visiflow.BLL.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Visiflow.Common.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Visiflow.Common.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Visiflow.DAL.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.Visiflow.DAL.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.Backend.Facade.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.Common.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.DomainRepository.Common.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.DomainRepository.LoanReview.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.DomainRepository.OCROrder.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.DomainRepository.ScreenConfigurations.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.ElasticSearch.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.Model.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.VLR.OCRDataMining.Backend.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.OCRFileOrder.Backend.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.VLR.OCRFileOrder.Backend.dll	MSIL_MSVC14_X86_64	Windows X86_64	20210819124611
Visionet.VLR.OCRFileOrder.Backend.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.OCRFileOrder.Backend.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.Web.Infrastructure.dll	MSIL_MSVC14_X86	Windows	20210819124611
Visionet.VLR.Web.Presentation.Utilities.dll	MSIL_MSVC14_X86	Windows	20210819124611

Module Name	Compiler	Operating Environment	Engine Version
Visionnet.VLR.LabelPrinting.dll	MSIL_MSVC14_X86	Windows	20210819124611
VLR.ClassLibrary.dll	MSIL_MSVC14_X86	Windows	20210819124611
VLR.ClassLibrary.dll	MSIL_MSVC14_X86	Windows	20210819124611
VLR.ClassLibrary.XmlSerializers.dll	MSIL_MSVC11_X86	Windows	20210819124611
VLR.Common.DAL.dll	MSIL_MSVC14_X86	Windows	20210819124611
VLR.Common.DAL.dll	MSIL_MSVC14_X86	Windows	20210819124611
VLR.Common.dll	MSIL_MSVC14_X86	Windows	20210819124611
VLR.Common.dll	MSIL_MSVC14_X86	Windows	20210819124611
VLRCache.dll	MSIL_MSVC14_X86	Windows	20210819124611
VLRCache.dll	MSIL_MSVC14_X86	Windows	20210819124611
VSIEncompassConnect.dll	MSIL_MSVC14_X86	Windows	20210819124611
VSIImportServiceLib.dll	MSIL_MSVC14_X86	Windows	20210819124611
WebDav.dll	MSIL_MSVC8_X86	Windows	20210819124611
WebDav.dll	MSIL_MSVC8_X86	Windows	20210819124611
Z.BulkOperations.dll	MSIL_MSVC14_X86	Windows	20210819124611

Flaw Types by Severity and Category

	Static Scan Security Quality Score = 57 (+1) from prior scan		
Very High	**0**		
High	**0**		
Medium	**241 (-14)**		
CRLF Injection	15		
Credentials Management	(-1)		
Cross-Site Scripting (XSS)	17 (-1)		
Cryptographic Issues	77 (-12)		
Directory Traversal	117		
Information Leakage	3		

	Static Scan Security Quality Score = 57 (+1) from prior scan		
Insufficient Input Validation	11		
Time and State	1		
Low	**154 (-6)**		
Cryptographic Issues	11 (-5)		
Information Leakage	35 (-1)		
Insufficient Input Validation	108		
Very Low	**0**		
Informational	**14 (-5)**		
Code Quality	14 (-5)		
Total	**409 (-25)**		

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	VL1	Did not pass
(VL3 + SCA) Min Analysis Score	70	57	Did not pass
(VL3 + SCA) Max Severity	High	Flaws found: 0	Passed

Software Composition Analysis Rules

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

Unsupported Frameworks

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

* 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

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 21 opportunities for this flaw, and 4 were successfully defended against using security best practices.
- The remaining 17 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 (241 flaws)

→ CRLF Injection(15 flaws)

Description

The acronym CRLF stands for "Carriage Return, Line Feed" and refers to the sequence of characters used to denote the end of a line of text. CRLF injection vulnerabilities occur when data enters an application from an untrusted source and is not properly validated before being used. For example, if an attacker is able to inject a CRLF into a log file, he could append falsified log entries, thereby misleading administrators or cover traces of the attack. If an attacker is able to inject CRLFs into an HTTP response header, he can use this ability to carry out other attacks such as cache poisoning. CRLF vulnerabilities primarily affect data integrity.

Recommendations

Apply robust input filtering for all user-supplied data, using centralized data validation routines when possible. Use output filters to sanitize all output derived from user-supplied input, replacing non-alphanumeric characters with their HTML entity equivalents.

Associated Flaws by CWE ID:

→ Improper Output Neutralization for Logs (CWE ID 117)(15 flaws)

Description

A function call could result in a log forging attack. Writing untrusted data into a log file allows an attacker to forge log entries or inject malicious content into log files. Corrupted log files can be used to cover an attacker's tracks or as a delivery mechanism for an attack on a log viewing or processing utility. For example, if a web administrator uses a browser-based utility to review logs, a cross-site scripting attack might be possible.

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

Recommendations

Avoid directly embedding user input in log files when possible. Sanitize untrusted data used to construct log entries by using a safe logging mechanism such as the OWASP ESAPI Logger, which will automatically remove unexpected carriage returns and line feeds and can be configured to use HTML entity encoding for non-alphanumeric data. Only write custom blacklisting code when absolutely necessary. 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
1071	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 271	
7	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 317	
1070	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 376	
23	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 422	
152	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 484	
21	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 532	
1069	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 591	
29	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 637	
1068	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 703	
10	25	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../logcentral.cs 749	
1186	-	44	vlr.classlibrary.dll	void FTPFiles(int) 14%	
1187	-	44	vlr.classlibrary.dll	void FTPFiles(int) 20%	
1189	-	44	vlr.classlibrary.dll	void FTPFiles(int) 73%	
1192	-	44	vlr.classlibrary.dll	void FTPFiles(int) 77%	
1182	-	34	vlr.classlibrary.dll	void WriteEventLog(string, System.Diagnostics.EventLogEntryType, int) 88%	

Cross-Site Scripting (XSS)(17 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)(17 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
1262	-	17	visionet.vlr.web.presentation.dll	string CreateUniqueFileName(string) 95%	
1522	-	25	visionet.vlr.web.presentation.dll	string GenrateFieldData(System.Collections.Generic.List<Visionet.VLR.OCRDataMining.Backend.DL.usp_Search_Detail_Fields_Data_Result>) 80%	
1898	-	22	visionet.vlr.web.presentation.dll	string GetFieldValue(string) 98%	
1899	-	26	visionet.vlr.web.presentation.dll	string GetFieldValue(string) 98%	
1513	-	22	visionet.vlr.web.presentation.dll	string GetLoanDocumentImage() 88%	
1520	-	22	visionet.vlr.web.presentation.dll	string GetLoanDocumentImage() 98%	
1433	-	21	visionet.vlr.web.presentation.dll	string GetLoanDocumentImage(string, string, string) 82%	
1434	-	21	visionet.vlr.web.presentation.dll	string GetLoanDocumentImage(string, string, string) 98%	
1412	-	20	visionet.vlr.web.presentation.dll	string GetQuery(Visionet.VLR.Model.TaskSaveTemplate[], string) 99%	
1383	-	23	visionet.vlr.web.presentation.dll	string GetTokenInfo() 80%	
1452	-	21	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%	
1469	-	21	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%	
1470	-	21	visionet.vlr.web.presentation.dll	string RenameDocumentSchemes(Visionet.VL	

	Flaw Id	Module #	Class #	Module	Location	Fix By
					R.OCRFileOrder.Backend.FileOrder, string, string, long, long) 81%	
	1471	-	21	visionet.vlr.web.presentation.dll	string RenameDocumentSchemes(Visionet.VLR.OCRFileOrder.Backend.FileOrder, string, string, long, long) 85%	
	1485	-	21	visionet.vlr.web.presentation.dll	string RenameDocumentSchemes(Visionet.VLR.OCRFileOrder.Backend.FileOrder, string, string, long, long) 92%	
	1897	-	23	visionet.vlr.web.presentation.dll	string SanitizeRedirectUrl(string) 98%	
NEW	1904	-	47	vlr.common.dll	void registerScript() 87%	

Cryptographic Issues(77 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:

Improper Certificate Validation (CWE ID 295)(12 flaws)

Description

The SSL Certificate Details provide information about the certificate associated with the HTTPS server. This information describes important attributes of the certificate and should be reviewed for the correctness of the contact information, whether it is self-signed, and how soon it will expire. The Supported Ciphers provides information about each available encryption scheme available for each of the possible SSL/TLS protocols.

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

Recommendations

The expiration date for certificates should be monitored. New certificates should be available before the expiration date in order to maintain continuity of service and avoid downtime or displaying certificate expiration errors in users' web browsers.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1154	-	3	document.splitting.service.exe#1.0.0.0	bool <CreateSession>b__3_0(object, System.Security.Cryptography.X509Certificates.X509Certificate, System.Security.Cryptography.X509Certificates.X509Chain, System.Net.Security.SslPolicyErrors) 50%	
1101	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 351	
1102	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 415	
1096	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 440	
1123	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 460	
1112	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 472	
1119	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 486	
1114	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 607	
1127	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 629	
1098	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 645	
1121	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 678	
1727	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 708	

→ Inadequate Encryption Strength (CWE ID 326)(2 flaws)

Description

Insufficiently strong encryption schemes may not adequately secure secret data from attackers. This can result from poor cipher selection, insufficient key size, or weak key selection.

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

Recommendations

Use a cryptographic algorithm that has been subject to public scrutiny. Follow security best practices when selecting key sizes and when generating key material.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1179	-	38	vlr.classlibrary.dll	void set_KeySizeBits(int) 28%	
1178	-	38	vlr.classlibrary.dll	void set_KeySizeBytes(int) 33%	

Insufficient Entropy (CWE ID 331)(16 flaws)

Description

Standard random number generators do not provide a sufficient amount of entropy when used for security purposes. Attackers can brute force the output of pseudorandom number generators such as rand().

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

Recommendations

If this random number is used where security is a concern, such as generating a session identifier or cryptographic key, use a trusted cryptographic random number generator instead.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
167	4	-	visionet.orderplacement.service.dll	systems working/.../common.cs 493	
155	4	-	visionet.orderplacement.service.dll	systems working/.../common.cs 496	
792	13	-	visionet.vlr.web.presentation.utilities.dll	.../dashboardcontrolshelper.cshtml 44	
607	13	-	visionet.vlr.web.presentation.utilities.dll	.../dashboardcontrolshelper.cshtml 165	
558	13	-	visionet.vlr.web.presentation.utilities.dll	.../dashboardcontrolshelper.cshtml 234	
586	13	-	visionet.vlr.web.presentation.utilities.dll	.../dashboardcontrolshelper.cshtml 398	
1405	-	18	visionet.vlr.web.presentation.dll	FileUploadJsonResult FileUpload_Click(System.Web.HttpPostedFileBase) 58%	
163	27	-	visionet.orderplacement.service.dll	.../ocrorderplacement.cs 375	
705	33	-	visionet.vlr.web.presentation.utilities.dll	.../screenhtmlhelper.cshtml 268	
560	33	-	visionet.vlr.web.presentation.utilities.dll	.../screenhtmlhelper.cshtml 389	
557	33	-	visionet.vlr.web.presentation.utilities.dll	.../screenhtmlhelper.cshtml 816	
820	34	-	visionet.vlr.web.presentation.utilities.dll	.../screenhtmlhelperportal.cshtml 503	
522	34	-	visionet.vlr.web.presentation.utilities.dll	.../screenhtmlhelperportal.cshtml 615	
1216	-	9	visionet.vlr.common	string GetRandomValue() 96%	

Flaw Id	Module #	Class #	Module	Location	Fix By
			.dll		
1388	-	18	visionet.vlr.web.presentation.dll	System.Web.Mvc.ActionResult DocumentUpload(System.Web.HttpPostedFileBase) 60%	
1361	-	23	visionet.vlr.web.presentation.dll	System.Web.Mvc.ActionResult LoginValidation() 45%	

Use of Hard-coded Cryptographic Key (CWE ID 321)(10 flaws)

Description

A method uses a hard-coded cryptographic key that may compromise system security in a way that cannot be easily remedied. The use of a hard-coded key significantly increases the possibility that encrypted data may be recovered. Moreover, the key cannot be changed without patching the software. If a hard-coded key is compromised in a commercial product, all deployed instances may be vulnerable to attack.

Effort to Fix: 4 - Simple design error. Requires redesign and up to 5 days to fix.

Recommendations

Store encryption keys out-of-band from the application code. Follow best practices for protecting keys stored in locations such as configuration or properties files.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1213	-	8	visionet.vlr.common.dll	byte[] DecryptAndGetFileContents(string) 67%	
1211	-	8	visionet.vlr.common.dll	byte[] DecryptFileStream(System.IO.Stream) 70%	
1076	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 108	
93	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 169	
92	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 222	
91	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 349	
95	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 396	
1209	-	8	visionet.vlr.common.dll	string EncryptAndSaveFile(string, string) 59%	
1588	-	8	visionet.vlr.common.dll	string EncryptAndSaveFile(System.IO.Stream, string) 59%	

Flaw Id	Module #	Class #	Module	Location	Fix By
1207	-	8	visionet.vlr.common .dll	string EncryptAndSaveFile(System.IO.Stream, string, bool) 60%	

→ Use of RSA Algorithm without OAEP (CWE ID 780)(2 flaws)

Description

The software uses the RSA algorithm but does not incorporate Optimal Asymmetric Encryption Padding (OAEP), which might weaken the encryption.

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

Recommendations

Use OAEP padding scheme when using RSA algorithm for encryption/decryption.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1181	-	36	vlr.classlibrary.dll	Data DecryptPrivate(Data) 95%	
1180	-	36	vlr.classlibrary.dll	Data EncryptPrivate(Data) 32%	

→ Use of a Broken or Risky Cryptographic Algorithm (CWE ID 327)(35 flaws)

Description

The use of a broken or risky cryptographic algorithm is an unnecessary risk that may result in the disclosure of sensitive information.

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
1212	-	8	visionet.vlr.common .dll	byte[] DecryptAndGetFileContents(string) 58%	
1210	-	8	visionet.vlr.common .dll	byte[] DecryptFileStream(System.IO.Stream) 60%	
1141	3	-	visionet.outboundintegration.service.exe #1.0.0.0	projects/.../common/common.cs 161	
1853	18	-	visionet.email.dll	.../common/encryptor.vb 7	
151	28	-	visionet.orderplacement.service.dll	.../orderplacement.svc.cs 1499	
166	28	-	visionet.orderplacement.service.dll	.../orderplacement.svc.cs 1570	

Flaw Id	Module #	Class #	Module	Location	Fix By
108	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 28	
106	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 51	
1087	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 104	
100	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 166	
90	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 219	
107	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 346	
101	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 393	
1586	-	8	visionet.vlr.common.dll	string Decrypt(string) 0%	
1671	-	45	vlr.common.dll	string Decrypt(string) 63%	
1220	-	29	visionet.vlr.web.presentation.dll	string Decrypt(string, bool) 26%	
1221	-	29	visionet.vlr.web.presentation.dll	string Decrypt(string, bool) 54%	
1222	-	29	visionet.vlr.web.presentation.dll	string Decrypt(string, bool) 64%	
1585	-	8	visionet.vlr.common.dll	string Encrypt(string) 0%	
1670	-	45	vlr.common.dll	string Encrypt(string) 68%	
1217	-	29	visionet.vlr.web.presentation.dll	string Encrypt(string, bool) 28%	
1218	-	29	visionet.vlr.web.presentation.dll	string Encrypt(string, bool) 55%	
1219	-	29	visionet.vlr.web.presentation.dll	string Encrypt(string, bool) 65%	
1208	-	8	visionet.vlr.common.dll	string EncryptAndSaveFile(string, string) 51%	
1587	-	8	visionet.vlr.common.dll	string EncryptAndSaveFile(System.IO.Stream, string) 51%	
1589	-	8	visionet.vlr.common.dll	string EncryptAndSaveFile(System.IO.Stream, string, bool) 51%	
1582	-	16	visionet.vlr.web.presentation.dll	System.Security.Cryptography.SymmetricAlgorithm GetSymmetricAlgo(System.Security.Cryptography.Xml.EncryptionMethod, byte[]) 64%	
1184	-	46	vlr.common.dll	void !cctor() 16%	

Flaw Id	Module #	Class #	Module	Location	Fix By
1183	-	32	vlr.classlibrary.dll	void !cctor() 16%	
1683	-	42	vlr.classlibrary.dll	void !cctor() 16%	
1173	-	37	vlr.classlibrary.dll	void !ctor(Provider) 55%	
1176	-	37	vlr.classlibrary.dll	void !ctor(Provider) 65%	
1174	-	38	vlr.classlibrary.dll	void !ctor(Provider, bool) 29%	
1175	-	38	vlr.classlibrary.dll	void !ctor(Provider, bool) 40%	
1177	-	38	vlr.classlibrary.dll	void !ctor(Provider, bool) 60%	

Directory Traversal(117 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)(117 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
1193	-	44	vlr.classlibrary.dll	bool FileExists(string) 10%	
1150	-	5	document.splitting.service.exe#1.0.0.0	bool MergeAndBookMarkPdfFiles(string[], string, string[], ref System.Collections.Generic.List<Common.FinalPackageDetails> /*1*/) 15%	
1151	-	5	document.splitting.service.exe#1.0.0.0	bool MergeAndNestedBookMarkPdfFiles(string[], string, string[], ref System.Collections.Generic.List<Common.FinalPackageDetails> /*1*/) 12%	
1152	-	5	document.splitting.service.exe#1.0.0.0	bool MergeAndNestedBookMarkPdfFiles(System.Collections.Generic.List<BookMark>, string, ref System.Collections.Generic.List<Common.FinalPackageDetails> /*1*/) 13%	
1147	-	2	document.splitting.service.exe#1.0.0.0	bool MergeFiles(string, System.Collections.Generic.List<string>) 11%	
1153	-	5	document.splitting.service.exe#1.0.0.0	bool MergeFiles(string, System.Collections.Generic.List<string>) 13%	
1156	-	4	document.splitting.service.exe#1.0.0.0	bool SingleIndexingResultOrderSplit(GetIndexingResults_forSplitting_Result, string, string, int) 12%	
1157	-	4	document.splitting.service.exe#1.0.0.0	bool SingleIndexingResultOrderSplit(GetIndexingResults_forSplitting_Result, string, string, int) 12%	
1167	-	12	visionet.api.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%	
1486	-	21	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*/) 6%	
1132	3	-	visionet.outboundintegration.service.exe#1.0.0.0	projects/.../common/common.cs 59	
1135	3	-	visionet.outboundintegration.service.exe#1.0.0.0	projects/.../common/common.cs 61	
1136	3	-	visionet.outboundintegration.service.exe	projects/.../common/common.cs 101	

Flaw Id	Module #	Class #	Module	Location	Fix By
			#1.0.0.0		
570	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 423	
533	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 868	
675	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1239	
889	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1245	
657	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1253	
815	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1291	
519	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1312	
899	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1313	
781	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1313	
692	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1314	
398	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1363	
649	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1369	
853	7	-	visionet.vlr.web.infrastructure.dll	.../commonservicehelper.cs 1382	
299	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 64	
1613	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 368	
304	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 372	
1610	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 514	
1609	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 580	
1612	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/vision	.../fileconverter.cs 657	

Flaw Id	Module #	Class #	Module	Location	Fix By
			et.fileconverter.lib.dll		
1611	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 757	
1608	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 828	
279	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 1145	
289	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 1157	
303	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 402	
283	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 404	
288	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 590	
296	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 590	
282	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 624	
285	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 860	
297	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 876	
306	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgen	.../ftplogcontroller.cs 876	

Flaw Id	Module #	Class #	Module	Location	Fix By
			erator.lib.dll		
290	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 878	
295	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 884	
300	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 884	
294	20	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.npfloanfoldersgenerator.lib.dll	.../ftplogcontroller.cs 913	
1297	-	12	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%	
473	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 258	
427	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 261	
696	23	-	visionet.vlr.web.infrastructure.dll	.../loandocumentshelper.cs 553	
654	24	-	visionet.vlr.web.infrastructure.dll	.../loanreviewhelper.cs 202	
445	24	-	visionet.vlr.web.infrastructure.dll	.../loanreviewhelper.cs 208	
539	24	-	visionet.vlr.web.infrastructure.dll	.../loanreviewhelper.cs 221	
908	24	-	visionet.vlr.web.infrastructure.dll	.../loanreviewhelper.cs 227	
725	24	-	visionet.vlr.web.infrastructure.dll	.../loanreviewhelper.cs 259	
907	24	-	visionet.vlr.web.infrastructure.dll	.../loanreviewhelper.cs 265	
796	24	-	visionet.vlr.web.infrastructure.dll	.../loanreviewhelper.cs 278	
514	24	-	visionet.vlr.web.infrastructure.dll	.../loanreviewhelper.cs 284	
858	24	-	visionet.vlr.web.infrastructure.dll	.../loanreviewhelper.cs 1040	
160	29	-	visionet.orderplacement.service.dll/bytescoutocrextraction.dll	.../ocr engine/pdfextractor.vb 204	

Flaw Id	Module #	Class #	Module	Location	Fix By
1129	29	-	visionet.orderplacement.service.dll/bytescoutocrextraction.dll	.../ocr engine/pdfextractor.vb 268	
165	30	-	visionet.orderplacement.service.dll	systems working/.../pdfutil.cs 36	
1139	32	-	visionet.outboundintegration.service.exe#1.0.0.0	.../responsexmlpackage.cs 1433	
1131	35	-	visionet.outboundintegration.service.exe#1.0.0.0	.../searchsummarysheet.cs 27	
1138	35	-	visionet.outboundintegration.service.exe#1.0.0.0	.../searchsummarysheet.cs 139	
1134	35	-	visionet.outboundintegration.service.exe#1.0.0.0	.../searchsummarysheet.cs 149	
1616	-	48	vlr.common.dll	string GenerateReport(ref System.Web.UI.Page /*1*/, CrystalDecisions.CrystalReports.Engine.ReportDocument, string) 53%	
1617	-	48	vlr.common.dll	string GenerateReport(ref System.Web.UI.Page /*1*/, CrystalDecisions.CrystalReports.Engine.ReportDocument, string) 73%	
1149	-	2	document.splitting.service.exe#1.0.0.0	string GetMergeDataCaptureIssueListString(string) 48%	
1148	-	2	document.splitting.service.exe#1.0.0.0	string GetMergeDataCaptureString(string) 48%	
1436	-	21	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) 33%	
1451	-	21	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) 67%	
1450	-	21	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) 67%	

Flaw Id	Module #	Class #	Module	Location	Fix By
1466	-	21	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) 72%	
1468	-	21	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) 77%	
1467	-	21	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) 77%	
1561	-	40	vlr.classlibrary.dll	System.Collections.ArrayList ValidateXmlFileFormat(string, int, ref int) 12%	
1546	-	40	vlr.classlibrary.dll	System.Data.DataSet ReadDataFromExcelFile(string, int) 7%	
1547	-	40	vlr.classlibrary.dll	System.Data.DataSet ReadDataFromExcelFileCustom(string, int, ref string /*1*/) 6%	
1160	-	4	document.splitting.service.exe#1.0.0.0	System.IO.FileStream GetDestinationFileStream(string, bool) 96%	
1158	-	4	document.splitting.service.exe#1.0.0.0	System.IO.MemoryStream GetSourceFileStream(string) 28%	
1511	-	21	visionet.vlr.web.presentation.dll	System.Web.Mvc.ActionResult DownloadFileBytesFromVLRDataPath() 71%	
1489	-	21	visionet.vlr.web.presentation.dll	System.Web.Mvc.ActionResult generateCheckListReportForTask(string, string, string, string) 91%	
1389	-	18	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult DeleteDocument(int, string, int, int) 33%	
1404	-	18	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult DeleteDocument(int, string, int, int) 76%	
1143	-	1	document.splitting.service.exe#1.0.0.0	void AddFileToZip(string, string, string, System.Nullable<int>, ref string /*1*/, ref string /*1*/) 33%	
1144	-	1	document.splitting.service.exe#1.0.0.0	void AddFileToZip(string, string, string, System.Nullable<int>, ref string /*1*/, ref string /*1*/) 37%	

Flaw Id	Module #	Class #	Module	Location	Fix By
1145	-	1	document.splitting.service.exe#1.0.0.0	void AddFileToZip(string, string, string, System.Nullable<int>, ref string /*1*/, ref string /*1*/) 37%	
1146	-	1	document.splitting.service.exe#1.0.0.0	void AddFileToZip(string, string, string, System.Nullable<int>, ref string /*1*/, ref string /*1*/) 53%	
1161	-	4	document.splitting.service.exe#1.0.0.0	void DeleteFiles(System.Collections.Generic.List<string>) 24%	
1169	-	12	visionet.api.dll/visionet.vlr.ocrfileorder.backend.dll	void DeleteFiles(System.Collections.Generic.List<System.Tuple<long, string, int>>) 30%	
1164	-	10	visionet.api.dll/visionet.vlr.ocrfileorder.backend.dll	void DeleteMergingDocuments(string, System.Collections.Generic.List<DocumentsToMerge_Result>) 69%	
1172	-	12	visionet.api.dll/visionet.vlr.ocrfileorder.backend.dll	void DeleteMergingDocuments(string, System.Collections.Generic.List<Get_DocumentsListForMerging_Result>) 67%	
1166	-	11	visionet.api.dll/visionet.vlr.ocrfileorder.backend.dll	void DeleteMergingDocuments(string, System.Collections.Generic.List<SP_GetDocumentsToMergeForLenderLoanNumber_Result>) 69%	
1168	-	12	visionet.api.dll/visionet.vlr.ocrfileorder.backend.dll	void DeleteParentIndexedFile(long, string, int) 12%	
1188	-	44	vlr.classlibrary.dll	void FTPFiles(int) 70%	
1190	-	44	vlr.classlibrary.dll	void FTPFiles(int) 74%	
1191	-	44	vlr.classlibrary.dll	void FTPFiles(int) 74%	
1159	-	4	document.splitting.service.exe#1.0.0.0	void GetSourceFileStream(ref System.IO.MemoryStream /*1*/, string) 24%	
1163	-	10	visionet.api.dll/visionet.vlr.ocrfileorder.backend.dll	void MergeDocument(System.Collections.Generic.List<DocumentsToMerge_Result>, IDX_AutoMergeDocuments_Result, string, int) 25%	
1171	-	12	visionet.api.dll/visionet.vlr.ocrfileorder.backend.dll	void MergeDocument(System.Collections.Generic.List<Get_DocumentsListForMerging_Result>, string, int, string, int[], System.Nullable<int>) 30%	
1165	-	11	visionet.api.dll/visionet.vlr.ocrfileorder.backend.dll	void MergeDocument(System.Collections.Generic.List<SP_GetDocumentsToMergeForLenderLoanNumber_Result>, IDX_AutoMergeDocumentsForLenderLoanNumber_Result, string, int) 28%	
1155	-	4	document.splitting.service.exe#1.0.0.0	void PerformSplittingV2(ProcessOrderSplitting_Result) 31%	
1170	-	12	visionet.api.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*/,	

Flaw Id	Module #	Class #	Module	Location	Fix By
				int) 27%	
1583	-	14	visionet.vlr.web.presentation.dll	void SendFileInChunks(ref System.Web.Mvc.ControllerContext /*1*/) 10%	
1584	-	14	visionet.vlr.web.presentation.dll	void SendGZipCompressedFileInChunks(ref System.Web.Mvc.ControllerContext /*1*/) 61%	
1185	-	43	vlr.classlibrary.dll	void UploadFile(string, string, bool) 53%	
1544	-	40	vlr.classlibrary.dll	void ValidateCustomSchema(string, System.Data.DataTable, System.Collections.ArrayList, System.Data.DataTable, ref string /*1*/, int, ref string /*1*/) 1%	
1545	-	40	vlr.classlibrary.dll	void ValidateSchema(string, System.Data.DataTable, System.Collections.ArrayList, System.Data.DataTable, ref string /*1*/, int, ref string /*1*/) 0%	
1194	-	44	vlr.classlibrary.dll	void WriteDataIntoFile(string, string) 29%	
1195	-	44	vlr.classlibrary.dll	void WriteDataIntoFile(string, string) 45%	
149	42	-	visionet.orderplacement.service.dll	systems working/.../xmlutil.cs 312	
153	42	-	visionet.orderplacement.service.dll	systems working/.../xmlutil.cs 457	
154	42	-	visionet.orderplacement.service.dll	systems working/.../xmlutil.cs 1103	
162	42	-	visionet.orderplacement.service.dll	systems working/.../xmlutil.cs 1126	

Information Leakage(3 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:

→ Improper Restriction of XML External Entity Reference (CWE ID 611)(1 flaw)

Description

The product processes an XML document that can contain XML entities with URLs that resolve to documents outside of the intended sphere of control, causing the product to embed incorrect documents into its output. By default, the XML entity resolver will attempt to resolve and retrieve external references. If attacker-controlled XML can be submitted to one of these functions, then the attacker could gain access to information about an internal network, local filesystem, or other sensitive data. This is known as an XML eXternal Entity (XXE) attack.

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

Recommendations

Configure the XML parser to disable external entity resolution.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1581	-	15	visionet.vlr.web.presentation.dll	void LoadXml(string) 82%	

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

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
158	4	-	visionet.orderplacement.service.dll	systems working/.../common.cs 101	
1355	-	23	visionet.vlr.web.presentation.dll	Models.CaptchaResponse ValidateCaptcha(string) 98%	

→ Insufficient Input Validation(11 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:

→ Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection') (CWE ID 90)(9 flaws)

Description

The software does not sufficiently sanitize special elements that are used in LDAP queries or responses, allowing attackers to modify the syntax, contents, or commands of the LDAP query before it is executed.

Effort to Fix: 3 - Complex implementation error. Fix is approx. 51-500 lines of code. Up to 5 days 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
1565	-	33	vlr.classlibrary.dll	System.Data.DataTable GetActiveDirectoryGroups(string) 55%	
1566	-	33	vlr.classlibrary.dll	System.Data.DataTable GetActiveDirectoryGroups(string) 57%	
1226	-	33	vlr.classlibrary.dll	UserInfo GetActiveDirectoryUserInfo(string, string) 31%	
1227	-	33	vlr.classlibrary.dll	UserInfo GetActiveDirectoryUserInfo(string, string) 35%	
1590	-	33	vlr.classlibrary.dll	void GetActiveDirectoryUsersNew(ref System.Data.DataTable /*1*/, string, string, string, string, string) 11%	
1591	-	33	vlr.classlibrary.dll	void GetActiveDirectoryUsersNew(ref System.Data.DataTable /*1*/, string, string, string, string, string) 12%	
1592	-	33	vlr.classlibrary.dll	void GetActiveDirectoryUsersNew(ref System.Data.DataTable /*1*/, string, string, string, string, string) 25%	

Flaw Id	Module #	Class #	Module	Location	Fix By
1593	-	33	vlr.classlibrary.dll	void GetActiveDirectoryUsersNew(ref System.Data.DataTable /*1*/, string, string, string, string, string) 28%	
1594	-	33	vlr.classlibrary.dll	void GetActiveDirectoryUsersNew(ref System.Data.DataTable /*1*/, string, string, string, string, string) 83%	

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
1896	-	23	visionet.vlr.web.presentation.dll	System.Web.Mvc.ActionResult RedirectToAction() 99%	
1435	-	21	visionet.vlr.web.presentation.dll	void ViewLoanDocumentImage() 92%	

Time and State(1 flaw)

Description

Time and State flaws are related to unexpected interactions between threads, processes, time, and information. These interactions happen through shared state: semaphores, variables, the filesystem, and basically anything that can store information. Vulnerabilities occur when there is a discrepancy between the programmer's assumption of how a program executes and what happens in reality.

State issues result from improper management or invalid assumptions about system state, such as assuming mutable objects are immutable. Though these conditions are less commonly exploited by attackers, state issues can lead to unpredictable or undefined application behavior.

Recommendations

Limit the interleaving of operations on resources from multiple processes. Use locking mechanisms to protect resources effectively. Follow best practices with respect to mutable objects and internal references. Pay close attention to asynchronous actions in processes and make copious use of sanity checks in systems that may be subject to synchronization errors.

Associated Flaws by CWE ID:

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

Description

Creating and using insecure temporary files can leave application and system data vulnerable to attack. In particular, file names created by the tmpnam family of functions can be easily guessed by an attacker. If an attacker can predict the filename and create a malicious collision, he may be able to manipulate the behavior of the application.

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

Recommendations

Ensure that unpredictable names are used for temporary files and that files are created in a secure directory with appropriate permissions. Using mkstemp() is a reasonably safe way to create temporary files. It will attempt to create and open a unique file based on a filename template provided by the user, combined with a series of randomly generated characters. Note that mkstemp() is safe if only the descriptor is used and the returned filename is not used in a subsequent function call with extra privileges. Using mkstemp() does not completely eliminate race conditions but does provide better protection than other methods.

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1142	-	7	visionet.api.dll/visionet.api.common.dll/kofax.omnipagecsdk.objects.dll	int TestRule(string) 5%	

Low (154 flaws)

Cryptographic Issues(11 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)(11 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
1233	-	8	visionet.vlr.common.dll	byte[] DecryptAndGetFileContents(string) 70%	
1232	-	8	visionet.vlr.common.dll	byte[] DecryptFileStream(System.IO.Stream) 73%	
1073	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 109	
99	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 170	
94	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 223	
111	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 350	
97	37	-	document.splitting.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 397	
1231	-	8	visionet.vlr.common.dll	string EncryptAndSaveFile(string, string) 62%	
1229	-	8	visionet.vlr.common.dll	string EncryptAndSaveFile(System.IO.Stream, string) 62%	
1230	-	8	visionet.vlr.common.dll	string EncryptAndSaveFile(System.IO.Stream, string, bool) 63%	
1137	40	-	visionet.outboundintegration.service.exe#1.0.0.0/vsiencompassconnect.dll	.../symmetriccryptographer.cs 153	

Information Leakage(35 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)(13 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
105	10	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.logging.dll	.../customlogging.cs 129	
103	10	-	visionet.npfloanfold	.../customlogging.cs 129	

	Flaw Id	Module #	Class #	Module	Location	Fix By
				ersgenerator.service.exe#1.0.0.0/visionet.logging.dll		
	1462	-	17	visionet.vlr.web.presentation.dll	string DDL_SUBTeamTaskAssociation(string) 98%	
	1418	-	17	visionet.vlr.web.presentation.dll	string GetAttachedRulesByEvent(string, string, string, string, bool) 99%	
	1474	-	17	visionet.vlr.web.presentation.dll	string GetCalendarConfiguration() 98%	
	1420	-	17	visionet.vlr.web.presentation.dll	string GetClientRuleText(string) 97%	
	1556	-	20	visionet.vlr.web.presentation.dll	string GetExceptionFieldMapping(Visionet.VLR.Model.DataFilter) 99%	
	1456	-	17	visionet.vlr.web.presentation.dll	string GetReassignUsers(string, string, string, string, string, string, string) 98%	
	1419	-	17	visionet.vlr.web.presentation.dll	string GetRuleText(string) 97%	
	1415	-	17	visionet.vlr.web.presentation.dll	string GetTaskFieldMapping(Visionet.VLR.Model.DataFilter) 99%	
	1463	-	17	visionet.vlr.web.presentation.dll	string OnChangeCustomObjectsDDL(string, string) 98%	
NEW	1903	-	47	vlr.common.dll	void registerScript() 87%	
NEW	1902	-	45	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
1202	-	43	vlr.classlibrary.dll	bool Login() 14%	
98	10	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/vision	.../customlogging.cs 129	

Flaw Id	Module #	Class #	Module	Location	Fix By
			et.logging.dll		
102	10	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.logging.dll	.../customlogging.cs 129	
317	38	-	visionet.email.dll	.../common/smtpemailer.vb 52	
323	38	-	visionet.email.dll	.../common/smtpemailer.vb 52	
1445	-	17	visionet.vlr.web.presentation.dll	string DataTableToJSONWithStringBuilder(System.Data.DataTable) 99%	
1517	-	23	visionet.vlr.web.presentation.dll	string GetTokenInfo() 72%	
1518	-	23	visionet.vlr.web.presentation.dll	string GetTokenInfo() 72%	
1531	-	18	visionet.vlr.web.presentation.dll	string GetUploadPath() 98%	
1595	-	21	visionet.vlr.web.presentation.dll	string GetUploadPath() 98%	
1476	-	17	visionet.vlr.web.presentation.dll	string InsertRegions(int, System.Collections.Generic.List<Models.Regions>) 99%	
1323	-	19	visionet.vlr.web.presentation.dll	System.Collections.Generic.List<FilterField> GetUserSavedFilters(int) 74%	
1324	-	19	visionet.vlr.web.presentation.dll	System.Collections.Generic.List<FilterField> GetUserSavedFilters(int) 74%	
1198	-	31	vlr.classlibrary.dll	void EmailLoanRequestReminder(VLR.Common.LoanRequestReminderInfo) 97%	
1199	-	31	vlr.classlibrary.dll	void EmailLoanRequestReminder(VLR.Common.LoanRequestReminderInfo) 97%	
1901	-	24	visionet.vlr.web.presentation.dll	void MoveNext() 48%	
1900	-	24	visionet.vlr.web.presentation.dll	void MoveNext() 48%	
1203	-	43	vlr.classlibrary.dll	void SendCommand(string) 74%	
1196	-	35	vlr.classlibrary.dll	void SendEmail(string, string, string, string, string, string, string, string, string, string[], (1 more parameter)) 98%	
1197	-	35	vlr.classlibrary.dll	void SendEmail(string, string, string, string, string, string, string, string, string, string[], (1 more parameter)) 98%	
1200	-	31	vlr.classlibrary.dll	void SendLoanRequestEmail(VLR.Common.LoanRequest) 97%	
1201	-	31	vlr.classlibrary.dll	void SendLoanRequestEmail(VLR.Common.LoanRequest) 97%	

→ Insufficient Input Validation(108 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)(108 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
177	1	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionfilter.cs 11	
205	1	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionfilter.cs 12	
213	1	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionfilter.cs 13	
240	1	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionfilter.cs 14	
185	1	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionfilter.cs 15	
175	1	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionfilter.cs 16	
212	1	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionfilter.cs 17	
197	1	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionfilter.cs 18	
179	2	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionsubinfofilters.cs 11	
182	2	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionsubinfofilters.cs 13	
219	2	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionsubinfofilters.cs 14	
230	2	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionsubinfofilters.cs 15	
232	2	-	visionet.api.dll/visio	.../batchtransactionsubinfofilters.cs 16	

Flaw Id	Module #	Class #	Module	Location	Fix By
			net.api.common.dll		
233	2	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionsubinfofilters.cs 18	
236	2	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionsubinfofilters.cs 19	
195	2	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionsubinfofilters.cs 20	
178	2	-	visionet.api.dll/visionet.api.common.dll	.../batchtransactionsubinfofilters.cs 21	
200	6	-	visionet.api.dll/visionet.api.common.dll	.../models/commonmodel.cs 11	
251	6	-	visionet.api.dll/visionet.api.common.dll	.../models/commonmodel.cs 12	
214	14	-	visionet.api.dll/visionet.api.common.dll	.../models/dashboardfiltermodel.cs 24	
231	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 13	
250	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 14	
174	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 15	
183	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 16	
194	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 17	
235	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 18	
192	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 19	
229	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 22	
188	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 23	
203	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 24	
206	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 25	
186	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 26	
209	15	-	visionet.api.dll/visionet.api.common.dll	.../models/documentfilter.cs 27	
204	16	-	visionet.api.dll/visionet.api.common.dll	.../models/documentpagefilter.cs 10	
246	16	-	visionet.api.dll/visionet.api.common.dll	.../models/documentpagefilter.cs 11	
225	16	-	visionet.api.dll/visionet.api.common.dll	.../models/documentpagefilter.cs 12	
227	21	-	visionet.api.dll/visionet.api.common.dll	.../models/getnextbatchfilter.cs 11	
218	21	-	visionet.api.dll/visionet.api.common.dll	.../models/getnextbatchfilter.cs 12	

Flaw Id	Module #	Class #	Module	Location	Fix By
243	21	-	visionet.api.dll/visionet.api.common.dll	.../models/getnextbatchfilter.cs 13	
180	22	-	visionet.api.dll/visionet.api.common.dll	.../gettextfieldconfigurationbyfileorderidviewmodel.cs 12	
184	26	-	visionet.api.dll/visionet.api.common.dll	.../models/lookupmodel.cs 12	
226	31	-	visionet.api.dll/visionet.api.common.dll	.../models/rejectpagesmodel.cs 11	
190	31	-	visionet.api.dll/visionet.api.common.dll	.../models/rejectpagesmodel.cs 13	
196	31	-	visionet.api.dll/visionet.api.common.dll	.../models/rejectpagesmodel.cs 19	
217	31	-	visionet.api.dll/visionet.api.common.dll	.../models/rejectpagesmodel.cs 20	
238	31	-	visionet.api.dll/visionet.api.common.dll	.../models/rejectpagesmodel.cs 21	
237	31	-	visionet.api.dll/visionet.api.common.dll	.../models/rejectpagesmodel.cs 26	
193	31	-	visionet.api.dll/visionet.api.common.dll	.../models/rejectpagesmodel.cs 27	
191	31	-	visionet.api.dll/visionet.api.common.dll	.../models/rejectpagesmodel.cs 28	
224	31	-	visionet.api.dll/visionet.api.common.dll	.../models/rejectpagesmodel.cs 29	
1309	-	28	visionet.vlr.web.presentation.dll	string get_AllowDelete() 0%	
1308	-	28	visionet.vlr.web.presentation.dll	string get_AlterScreenURL() 0%	
1318	-	30	visionet.vlr.web.presentation.dll	string get_AssignmentInformation() 0%	
1352	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_AtCloseOrderNo() 0%	
1366	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_BatchName() 0%	
1351	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_ClientFilePath() 0%	
1367	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_ClientLoanNumber() 0%	
1273	-	27	visionet.vlr.web.presentation.dll	string get_Code_Translation_Tepmlate_Name() 0%	
1275	-	27	visionet.vlr.web.presentation.dll	string get_Column_Alias() 0%	
1317	-	30	visionet.vlr.web.presentation.dll	string get_CreateAssignments() 0%	

Flaw Id	Module #	Class #	Module	Location	Fix By
1321	-	30	visionet.vlr.web.presentation.dll	string get_CreateOutsourceListing() 0%	
1316	-	30	visionet.vlr.web.presentation.dll	string get_CreateTransmittal() 0%	
1276	-	27	visionet.vlr.web.presentation.dll	string get_CSVSeperator() 0%	
1343	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_CustomerName() 0%	
1338	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_CustomerOrderID() 0%	
1313	-	30	visionet.vlr.web.presentation.dll	string get_EmailAddress() 0%	
1278	-	28	visionet.vlr.web.presentation.dll	string get_Entity_Name() 0%	
1277	-	28	visionet.vlr.web.presentation.dll	string get_EntityId() 0%	
1320	-	30	visionet.vlr.web.presentation.dll	string get_EscalatedCalls() 0%	
1268	-	27	visionet.vlr.web.presentation.dll	string get_FileFieldName() 0%	
1339	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_FileName() 0%	
1340	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_FileNameAlias() 0%	
1306	-	28	visionet.vlr.web.presentation.dll	string get_FilterColumn_Name() 0%	
1307	-	28	visionet.vlr.web.presentation.dll	string get_FilterCoumnValue() 0%	
1311	-	30	visionet.vlr.web.presentation.dll	string get_FirstName() 0%	
1312	-	30	visionet.vlr.web.presentation.dll	string get_LastName() 0%	
1345	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_LoanNumber() 0%	
1315	-	30	visionet.vlr.web.presentation.dll	string get_LoginName() 0%	
1353	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_OPACode() 0%	
1341	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_PhysicalPath() 0%	
1314	-	30	visionet.vlr.web.pres	string get_Privilege() 0%	

Flaw Id	Module #	Class #	Module	Location	Fix By
			entation.dll		
1346	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_ProcessingMachineName() 0%	
1347	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_ProductCode() 0%	
1319	-	30	visionet.vlr.web.presentation.dll	string get_ReportAccess() 0%	
1349	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_SchemeName() 0%	
1348	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_SequenceID() 0%	
1274	-	27	visionet.vlr.web.presentation.dll	string get_Sheet_Name() 0%	
1342	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_SourceProject() 0%	
1350	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_SplittedStatus() 0%	
1269	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Datatype() 0%	
1270	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Default_Value() 0%	
1267	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Name() 0%	
1271	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Precesion() 0%	
1272	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Column_Scale() 0%	
1264	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Name() 0%	
1266	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Table_Description() 0%	
1265	-	27	visionet.vlr.web.presentation.dll	string get_Staging_Area_Table_Name() 0%	
1279	-	28	visionet.vlr.web.presentation.dll	string get_Table_Name() 0%	
1263	-	27	visionet.vlr.web.presentation.dll	string get_TemplateDesc() 0%	
1235	-	27	visionet.vlr.web.presentation.dll	string get_TemplateName() 0%	

Flaw Id	Module #	Class #	Module	Location	Fix By
1234	-	27	visionet.vlr.web.presentation.dll	string get_TemplateTypeName() 0%	
1354	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_TransactionId() 0%	
1344	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_TransmissionCode() 0%	
1310	-	30	visionet.vlr.web.presentation.dll	string get_UserName() 0%	
1322	-	13	visionet.vlr.web.presentation.dll/visionet.vlr.ocrfileorder.backend.dll	string get_VSIOrderID() 0%	
1492	-	17	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult DataImportTemplateSave(Models.DataImportModel) 0%	
1475	-	17	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult MaintainUserProfile(Models.UserMaintainProfileModel) 0%	
1441	-	17	visionet.vlr.web.presentation.dll	System.Web.Mvc.JsonResult SaveEntity(Models.EntityModel) 0%	

Very Low (0 flaws)

No flaws of this type were found

Info (14 flaws)

→ Code Quality(14 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)(14 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
1130	11	-	visionet.outboundintegration.service.exe#1.0.0.0/vsiencompassconnect.dll	temp/.../dal/dalbase.cs 46	
221	12	-	visionet.api.dll	.../dashboardcontroller.cs 99	
187	12	-	visionet.api.dll	.../dashboardcontroller.cs 247	
309	18	-	visionet.email.dll	.../common/encryptor.vb 32	
310	18	-	visionet.email.dll	.../common/encryptor.vb 48	
316	18	-	visionet.email.dll	.../common/encryptor.vb 61	
287	19	-	visionet.npfloanfoldersgenerator.service.exe#1.0.0.0/visionet.fileconverter.lib.dll	.../fileconverter.cs 842	
1	-	6	visionet.api.dll/visionet.api.common.dll/kofax.omnipagecsdk.argtypes.dll	int ROpenStreamCallBack(string, int, ref System.UIntPtr) 53%	
118	37	-	visionet.api.dll/visionet.logging.dll	.../securitymanager.cs 37	
1205	-	41	vlr.classlibrary.dll	void !ctor() 66%	
1206	-	39	vlr.classlibrary.dll	void !ctor() 66%	
1204	-	38	vlr.classlibrary.dll	void !ctor(Provider, bool) 50%	
1162	-	4	document.splitting.service.exe#1.0.0.0	void GetSourceFileStream(ref System.IO.MemoryStream /*1*/, string) 9%	
164	42	-	visionet.orderplacement.service.dll	systems working/.../xmlutil.cs 59	

Flaws in Common Modules

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

Module: VLR.ClassLibrary.dll Used by 13 executables; score impact: 3

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
vlr_classlibrary_dll.ImportUsersBLL					
GetActiveDirectoryGroups 57%	3	1	Insufficient Input Validation	90	Neutral
GetActiveDirectoryUsersNew 28%	3	1	Insufficient Input Validation	90	Neutral
GetActiveDirectoryGroups 55%	3	1	Insufficient Input Validation	90	Neutral
GetActiveDirectoryUsersNew 83%	3	1	Insufficient Input Validation	90	Neutral
GetActiveDirectoryUsersNew 11%	3	1	Insufficient Input Validation	90	Neutral
GetActiveDirectoryUserInfo 31%	3	1	Insufficient Input Validation	90	Neutral
GetActiveDirectoryUsersNew 12%	3	1	Insufficient Input Validation	90	Neutral
GetActiveDirectoryUserInfo 35%	3	1	Insufficient Input Validation	90	Neutral
GetActiveDirectoryUsersNew 25%	3	1	Insufficient Input Validation	90	Neutral
vlr_classlibrary_dll.cFTPService					
FTPFiles 14%	3	2	CRLF Injection	117	Likely
FTPFiles 73%	3	1	CRLF Injection	117	Likely
FTPFiles 77%	3	1	CRLF Injection	117	Likely
FTPFiles 20%	3	2	CRLF Injection	117	Likely
FileExists 10%	3	2	Directory Traversal	73	Neutral
WriteDataIntoFile 45%	3	2	Directory Traversal	73	Neutral
WriteDataIntoFile 29%	3	2	Directory Traversal	73	Neutral
FTPFiles 70%	3	2	Directory Traversal	73	Neutral
FTPFiles 74%	3	2	Directory Traversal	73	Neutral
vlr_classlibrary_dll.VLR.ClassLibrary.Encryption.Symmetric					
set_KeySizeBytes 33%	3	2	Cryptographic Issues	326	V.Likely
set_KeySizeBits 28%	3	2	Cryptographic Issues	326	V.Likely
!ctor 40%	3	2	Cryptographic Issues	327	Likely
!ctor 29%	3	2	Cryptographic Issues	327	Likely
!ctor 60%	3	2	Cryptographic Issues	327	Likely
!ctor 50%	0	2	Code Quality	404	Neutral
vlr_classlibrary_dll.VLR.TapeCracking.FileUploadHistoryBLL					
ValidateCustomSchema 1%	3	1	Directory Traversal	73	Neutral
ValidateSchema 0%	3	1	Directory Traversal	73	Neutral
ReadDataFromExcelFileCustom 6%	3	1	Directory Traversal	73	Neutral
ValidateXmlFileFormat 12%	3	1	Directory Traversal	73	Neutral
ReadDataFromExcelFile 7%	3	1	Directory Traversal	73	Neutral
vlr_classlibrary_dll.VLR.ClassLibrary.Encryption.Asymmetric					
DecryptPrivate 95%	3	2	Cryptographic Issues	780	Neutral
EncryptPrivate 32%	3	2	Cryptographic Issues	780	Neutral

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
vlr_classlibrary_dll.VLR.ClassLibrary.Encryption.Hash					
!ctor 65%	3	2	Cryptographic Issues	327	Likely
!ctor 55%	3	2	Cryptographic Issues	327	Likely
vlr_classlibrary_dll.cFTP					
UploadFile 53%	3	2	Directory Traversal	73	Neutral
Login 14%	2	2	Information Leakage	201	Unlikely
SendCommand 74%	2	2	Information Leakage	201	Unlikely
vlr_classlibrary_dll.Encryptor					
!cctor 16%	3	1	Cryptographic Issues	327	Likely
vlr_classlibrary_dll.VLR.ClassLibrary.CommonFunsBLL					
WriteEventLog 88%	3	2	CRLF Injection	117	Likely
vlr_classlibrary_dll.cEncryptor					
!cctor 16%	3	1	Cryptographic Issues	327	Likely
vlr_classlibrary_dll.AsyncEmailManager					
EmailLoanRequestReminder 97%	2	2	Information Leakage	201	Unlikely
SendLoanRequestEmail 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 66%	0	1	Code Quality	404	Neutral
vlr_classlibrary_dll.VLR.TapeCracking.cXLTemplateDAL					
!ctor 66%	0	1	Code Quality	404	Neutral

Module: Visionet.VLR.Web.Infrastructure.dll Used by 2 executables; score impact: 3

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
loandocumentshelper.cs 258	3	1	Directory Traversal	73	Neutral
loandocumentshelper.cs 261	3	1	Directory Traversal	73	Neutral
loandocumentshelper.cs 351	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 415	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 440	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 460	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 472	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 486	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 553	3	1	Directory Traversal	73	Neutral
loandocumentshelper.cs 607	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 629	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 645	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 678	3	1	Cryptographic Issues	295	Neutral
loandocumentshelper.cs 708	3	1	Cryptographic Issues	295	Neutral
commonservicehelper.cs 423	3	1	Directory Traversal	73	Neutral

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
commonservicehelper.cs 868	3	1	Directory Traversal	73	Neutral
commonservicehelper.cs 1239	3	1	Directory Traversal	73	Neutral
commonservicehelper.cs 1245	3	1	Directory Traversal	73	Neutral
commonservicehelper.cs 1253	3	1	Directory Traversal	73	Neutral
commonservicehelper.cs 1291	3	1	Directory Traversal	73	Neutral
commonservicehelper.cs 1312	3	1	Directory Traversal	73	Neutral
commonservicehelper.cs 1313	3	2	Directory Traversal	73	Neutral
commonservicehelper.cs 1314	3	1	Directory Traversal	73	Neutral
commonservicehelper.cs 1363	3	1	Directory Traversal	73	Neutral
commonservicehelper.cs 1369	3	1	Directory Traversal	73	Neutral
commonservicehelper.cs 1382	3	1	Directory Traversal	73	Neutral
loanreviewhelper.cs 202	3	1	Directory Traversal	73	Neutral
loanreviewhelper.cs 208	3	1	Directory Traversal	73	Neutral
loanreviewhelper.cs 221	3	1	Directory Traversal	73	Neutral
loanreviewhelper.cs 227	3	1	Directory Traversal	73	Neutral
loanreviewhelper.cs 259	3	1	Directory Traversal	73	Neutral
loanreviewhelper.cs 265	3	1	Directory Traversal	73	Neutral
loanreviewhelper.cs 278	3	1	Directory Traversal	73	Neutral
loanreviewhelper.cs 284	3	1	Directory Traversal	73	Neutral
loanreviewhelper.cs 1040	3	1	Directory Traversal	73	Neutral

Module: Visionet.Logging.dll Used by 59 executables; score impact: 2

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
securitymanager.cs 28	3	11	Cryptographic Issues	327	Likely
securitymanager.cs 51	3	11	Cryptographic Issues	327	Likely
securitymanager.cs 104	3	11	Cryptographic Issues	327	Likely
securitymanager.cs 108	3	11	Cryptographic Issues	321	Likely
securitymanager.cs 166	3	11	Cryptographic Issues	327	Likely
securitymanager.cs 169	3	11	Cryptographic Issues	321	Likely
securitymanager.cs 219	3	11	Cryptographic Issues	327	Likely
securitymanager.cs 222	3	11	Cryptographic Issues	321	Likely
securitymanager.cs 346	3	11	Cryptographic Issues	327	Likely
securitymanager.cs 349	3	11	Cryptographic Issues	321	Likely
securitymanager.cs 393	3	11	Cryptographic Issues	327	Likely
securitymanager.cs 396	3	11	Cryptographic Issues	321	Likely
securitymanager.cs 37	0	5	Code Quality	404	Neutral
securitymanager.cs 109	2	11	Cryptographic Issues	329	Neutral
securitymanager.cs 170	2	11	Cryptographic Issues	329	Neutral
securitymanager.cs 223	2	11	Cryptographic Issues	329	Neutral

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
securitymanager.cs 350	2	11	Cryptographic Issues	329	Neutral
securitymanager.cs 397	2	11	Cryptographic Issues	329	Neutral
logcentral.cs 271	3	9	CRLF Injection	117	Likely
logcentral.cs 317	3	4	CRLF Injection	117	Likely
logcentral.cs 376	3	9	CRLF Injection	117	Likely
logcentral.cs 422	3	4	CRLF Injection	117	Likely
logcentral.cs 484	3	9	CRLF Injection	117	Likely
logcentral.cs 532	3	4	CRLF Injection	117	Likely
logcentral.cs 591	3	9	CRLF Injection	117	Likely
logcentral.cs 637	3	4	CRLF Injection	117	Likely
logcentral.cs 703	3	9	CRLF Injection	117	Likely
logcentral.cs 749	3	4	CRLF Injection	117	Likely
customlogging.cs 129	2	2	Information Leakage	201	Unlikely
customlogging.cs 129	2	2	Information Leakage	209	Neutral

Module: Visionet.VLR.OCRFileOrder.Backend.dll Used by 17 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.BL.OCRFileOrderBL					
DeleteMergingDocuments 67%	3	3	Directory Traversal	73	Neutral
DeleteParentIndexedFile 12%	3	3	Directory Traversal	73	Neutral
RecreateParentIndexedFiles_ITextSharp 27%	3	3	Directory Traversal	73	Neutral
SplitFileITextSharp 56%	3	1	Directory Traversal	73	Neutral
DeleteFiles 30%	3	3	Directory Traversal	73	Neutral
SplitChildDocumentsList_ITextSharp 60%	3	3	Directory Traversal	73	Neutral
MergeDocument 30%	3	3	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_SequenceID 0%	2	1	Insufficient Input Validation	1174	Neutral
get_ClientLoanNumber 0%	2	1	Insufficient Input Validation	1174	Neutral
get_FileName 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_BatchName 0%	2	1	Insufficient Input Validation	1174	Neutral
get_FileNameAlias 0%	2	1	Insufficient Input Validation	1174	Neutral
get_ClientFilePath 0%	2	1	Insufficient Input Validation	1174	Neutral

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
get_SchemeName 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_CustomerOrderID 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	3	Directory Traversal	73	Neutral
DeleteMergingDocuments 69%	3	3	Directory Traversal	73	Neutral
visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.BL.AutoMergeForLenderLoanNumberBL					
DeleteMergingDocuments 69%	3	3	Directory Traversal	73	Neutral
MergeDocument 28%	3	3	Directory Traversal	73	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 0%	3	1	Cryptographic Issues	327	Likely
EncryptAndSaveFile 59%	3	1	Cryptographic Issues	321	Likely
EncryptAndSaveFile 59%	3	1	Cryptographic Issues	321	Likely
EncryptAndSaveFile 51%	3	1	Cryptographic Issues	327	Likely
Decrypt 0%	3	1	Cryptographic Issues	327	Likely
DecryptFileStream 60%	3	1	Cryptographic Issues	327	Likely
DecryptFileStream 70%	3	1	Cryptographic Issues	321	Likely
EncryptAndSaveFile 51%	3	1	Cryptographic Issues	327	Likely
DecryptAndGetFileContents 67%	3	1	Cryptographic Issues	321	Likely
DecryptAndGetFileContents 58%	3	1	Cryptographic Issues	327	Likely
EncryptAndSaveFile 51%	3	1	Cryptographic Issues	327	Likely
EncryptAndSaveFile 60%	3	1	Cryptographic Issues	321	Likely
DecryptAndGetFileContents 70%	2	1	Cryptographic Issues	329	Neutral
DecryptFileStream 73%	2	1	Cryptographic Issues	329	Neutral
EncryptAndSaveFile 62%	2	1	Cryptographic Issues	329	Neutral
EncryptAndSaveFile 63%	2	1	Cryptographic Issues	329	Neutral
EncryptAndSaveFile 62%	2	1	Cryptographic Issues	329	Neutral
visionet_vlr_common_dll.Visionet.VLR.Common.VLRConstants					
GetRandomValue 96%	3	1	Cryptographic Issues	331	Unlikely

Module: Visionet.API.Common.dll Used by 4 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
documentfilter.cs 13	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 14	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 15	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 16	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 17	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 18	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 19	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 22	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 23	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 24	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 25	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 26	2	1	Insufficient Input Validation	1174	Neutral
documentfilter.cs 27	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionsubinfofilters.cs 11	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionsubinfofilters.cs 13	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionsubinfofilters.cs 14	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionsubinfofilters.cs 15	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionsubinfofilters.cs 16	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionsubinfofilters.cs 18	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionsubinfofilters.cs 19	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionsubinfofilters.cs 20	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionsubinfofilters.cs 21	2	1	Insufficient Input Validation	1174	Neutral
rejectpagesmodel.cs 11	2	1	Insufficient Input Validation	1174	Neutral
rejectpagesmodel.cs 13	2	1	Insufficient Input Validation	1174	Neutral
rejectpagesmodel.cs 19	2	1	Insufficient Input Validation	1174	Neutral
rejectpagesmodel.cs 20	2	1	Insufficient Input Validation	1174	Neutral
rejectpagesmodel.cs 21	2	1	Insufficient Input Validation	1174	Neutral
rejectpagesmodel.cs 26	2	1	Insufficient Input Validation	1174	Neutral
rejectpagesmodel.cs 27	2	1	Insufficient Input Validation	1174	Neutral
rejectpagesmodel.cs 28	2	1	Insufficient Input Validation	1174	Neutral
rejectpagesmodel.cs 29	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionfilter.cs 11	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionfilter.cs 12	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionfilter.cs 13	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionfilter.cs 14	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionfilter.cs 15	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionfilter.cs 16	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionfilter.cs 17	2	1	Insufficient Input Validation	1174	Neutral
batchtransactionfilter.cs 18	2	1	Insufficient Input Validation	1174	Neutral
documentpagefilter.cs 10	2	1	Insufficient Input Validation	1174	Neutral

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
documentpagefilter.cs 11	2	1	Insufficient Input Validation	1174	Neutral
documentpagefilter.cs 12	2	1	Insufficient Input Validation	1174	Neutral
getnextbatchfilter.cs 11	2	1	Insufficient Input Validation	1174	Neutral
getnextbatchfilter.cs 12	2	1	Insufficient Input Validation	1174	Neutral
getnextbatchfilter.cs 13	2	1	Insufficient Input Validation	1174	Neutral
commonmodel.cs 11	2	1	Insufficient Input Validation	1174	Neutral
commonmodel.cs 12	2	1	Insufficient Input Validation	1174	Neutral
dashboardfiltermodel.cs 24	2	1	Insufficient Input Validation	1174	Neutral
gettextfieldconfigurationbyfileorderidviewmodel.cs 12	2	1	Insufficient Input Validation	1174	Neutral
lookupmodel.cs 12	2	1	Insufficient Input Validation	1174	Neutral

Module: Visionet.FileConverter.Lib.dll Used by 2 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
fileconverter.cs 64	3	1	Directory Traversal	73	Neutral
fileconverter.cs 368	3	1	Directory Traversal	73	Neutral
fileconverter.cs 372	3	1	Directory Traversal	73	Neutral
fileconverter.cs 514	3	1	Directory Traversal	73	Neutral
fileconverter.cs 580	3	1	Directory Traversal	73	Neutral
fileconverter.cs 657	3	1	Directory Traversal	73	Neutral
fileconverter.cs 757	3	1	Directory Traversal	73	Neutral
fileconverter.cs 828	3	1	Directory Traversal	73	Neutral
fileconverter.cs 1145	3	1	Directory Traversal	73	Neutral
fileconverter.cs 1157	3	1	Directory Traversal	73	Neutral
fileconverter.cs 842	0	1	Code Quality	404	Neutral

Module: VLR.Common.dll Used by 24 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
vlr_common_dll.VLR.Common.CommonLib					
Decrypt 63%	3	3	Cryptographic Issues	327	Likely
Encrypt 68%	3	3	Cryptographic Issues	327	Likely
ShowMessage 63%	2	4	Information Leakage	209	Neutral
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.Encryptor					
!cctor 16%	3	2	Cryptographic Issues	327	Likely
vlr_common_dll.VLR.Common.Utility					
registerScript 87%	2	2	Information Leakage	209	Neutral

Module: Visionet.EMail.dll Used by 15 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
encryptor.vb 7	3	2	Cryptographic Issues	327	Likely
encryptor.vb 32	0	2	Code Quality	404	Neutral
encryptor.vb 48	0	2	Code Quality	404	Neutral
encryptor.vb 61	0	2	Code Quality	404	Neutral
smtpemailer.vb 52	2	2	Information Leakage	201	Unlikely

Module: Kofax.OmniPageCSDK.Objects.dll Used by 6 executables; score impact: 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
kofax_omnipagecsdk_objects_dll.Kofax.OmniPageCSDK.Objects.DataRule					
TestRule 5%	3	2	Time and State	377	Neutral

Module: Kofax.OmniPageCSDK.ArgTypes.dll Used by 14 executables; score impact: < 1

Location	Severity	# Instances	Flaw Category	CWE ID	Exploitability
kofax_omnipagecsdk_argtypes_dll.Kofax.OmniPageCSDK.ArgTypes.IOStreamCB					
ROpenStreamCallBack 53%	0	2	Code Quality	404	Neutral

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:	1 Sep 2021 Static	Scan Name:	24 Aug 2021 Static
Completed:	9/1/21	Completed:	8/24/21
Score:	57	Score:	56

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 (20 flaws)

→ Credentials Management(1 flaw)

Associated Flaws by CWE ID:

→ Use of Hard-coded Password (CWE ID 259)(1 flaw)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
129	39	-	vlr.common.dll	.../vlr.common/sqlspconstants.vb 12824	

→ Cross-Site Scripting (XSS)(2 flaws)

Associated Flaws by CWE ID:

→ Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) (CWE ID 80)(2 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
648	8	-	vlr.common.dll	.../vlr.common/commonutility.vb 1953	
541	41	-	vlr.common.dll	systems working/.../utility.vb 1975	

→ Cryptographic Issues(15 flaws)

Associated Flaws by CWE ID:

→ Use of Hard-coded Cryptographic Key (CWE ID 321)(5 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1889	36	-	visionet.ftporderscontroller.service.exe#1.0.0.0/visionet.logging.dll	.../securitymanager.cs 108	

Flaw Id	Module #	Class #	Module	Location	Fix By
1881	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 169	
1883	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 222	
1882	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 349	
1880	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 396	

→ Use of a Broken or Risky Cryptographic Algorithm (CWE ID 327)(10 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
136	5	-	vlr.common.dll	.../vlr.common/commonlib.vb 769	
141	5	-	vlr.common.dll	.../vlr.common/commonlib.vb 800	
353	17	-	vlr.common.dll	.../vlr.common/encryptor.vb 7	
1894	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 28	
1891	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 51	
1879	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 104	
1885	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 166	
1892	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 219	
1886	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 346	
1895	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 393	

→ Directory Traversal(2 flaws)

Associated Flaws by CWE ID:

→ External Control of File Name or Path (CWE ID 73)(2 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
145	9	-	vlr.common.dll	.../vlr.common/creportmanager.vb 438	
132	9	-	vlr.common.dll	.../vlr.common/creportmanager.vb 449	

Low (8 flaws)

→ Cryptographic Issues(5 flaws)

Associated Flaws by CWE ID:

→ Generation of Predictable IV with CBC Mode (CWE ID 329)(5 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
1893	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 109	
1890	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 170	
1887	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 223	
1888	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 350	
1884	36	-	visionet.ftpordersco ntroller.service.exe# 1.0.0.0/visionet.logg ing.dll	.../securitymanager.cs 397	

→ Information Leakage(3 flaws)

Associated Flaws by CWE ID:

→ Generation of Error Message Containing Sensitive Information (CWE ID 209)(3 flaws)

Instances found via Static Scan

Flaw Id	Module #	Class #	Module	Location	Fix By
148	5	-	vlr.common.dll	.../vlr.common/commonlib.vb 80	
1108	8	-	vlr.common.dll	.../vlr.common/commonutility.vb 1953	
1105	41	-	vlr.common.dll	systems working/.../utility.vb 1975	

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
146	5	-	vlr.common.dll	.../vlr.common/commonlib.vb 767	
128	5	-	vlr.common.dll	.../vlr.common/commonlib.vb 799	
125	17	-	vlr.common.dll	.../vlr.common/encryptor.vb 30	
143	17	-	vlr.common.dll	.../vlr.common/encryptor.vb 46	
127	17	-	vlr.common.dll	.../vlr.common/encryptor.vb 59	

Appendix B: Referenced Source Files

Id	Filename	Path
1	batchtransactionfilter.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
2	batchtransactionsubinfofilters.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
3	common.cs	projects/pnq/source code dev/windows services/visionet.outboundintegration.service/visionet.outboundintegration.service/common/
4	common.cs	systems working/pnc indexing/source code dev/windows services/visionet.orderplacement.service/visionet.orderplacement.service/common/
5	commonlib.vb	systems working/pnc indexing/source code dev/visionet.vlr/vlr_backend/vlr.common/
6	commonmodel.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
7	commonservicehelper.cs	systems working/pnc indexing/source code dev/visionet.vlr/visionet.vlr.web.infrastructure/
8	commonutility.vb	systems working/pnc indexing/source code dev/visionet.vlr/vlr_backend/vlr.common/
9	creportmanager.vb	systems working/pnc indexing/source code dev/visionet.vlr/vlr_backend/vlr.common/
10	customlogging.cs	systems working/pnc indexing/source code dev/visionet.vlr/visionnet.vlr.logging/
11	dalbase.cs	temp/pnc indexing/visionet.outboundintegration.service/vsiencompassconnect/dal/
12	dashboardcontroller.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api/controllers/
13	dashboardcontrolshelper.cshtml	systems working/pnc indexing/source code dev/visionet.vlr/renderers/
14	dashboardfiltermodel.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
15	documentfilter.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
16	documentpagefilter.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
17	encryptor.vb	systems working/pnc indexing/source code dev/visionet.vlr/vlr_backend/vlr.common/
18	encryptor.vb	systems working/pnc indexing/source code dev/visionet.vlr/vlr_backend/visionet.email/common/
19	fileconverter.cs	systems working/pnc indexing/source code dev/windows services/visionet.npfloanfoldersgenerator.service/visionet.fileconverter.lib/
20	ftplogcontroller.cs	systems working/pnc indexing/source code dev/windows services/visionet.npfloanfoldersgenerator.service/visionet.npfloanfoldersgenerator.lib/
21	getnextbatchfilter.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
22	gettextfieldconfigurationbyfileorderidviewmodel.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
23	loandocumentshelper.cs	systems working/pnc indexing/source code dev/visionet.vlr/visionet.vlr.web.infrastructure/
24	loanreviewhelper.cs	systems working/pnc indexing/source code dev/visionet.vlr/visionet.vlr.web.infrastructure/

Id	Filename	Path
25	logcentral.cs	systems working/pnc indexing/source code dev/visionet.vlr/visionnet.vlr.logging/
26	lookupmodel.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
27	ocrorderplacement.cs	systems working/pnc indexing/source code dev/windows services/visionet.orderplacement.service/visionet.orderplacement.service/business classes/
28	orderplacement.svc.cs	systems working/pnc indexing/source code dev/windows services/visionet.orderplacement.service/visionet.orderplacement.service/
29	pdfextractor.vb	systems working/pnc indexing/source code dev/windows services/visionet.orderplacement.service/bytescoutocrextraction/ocr engine/
30	pdfutil.cs	systems working/pnc indexing/source code dev/windows services/visionet.orderplacement.service/visionet.orderplacement.service/common/
31	rejectpagesmodel.cs	systems working/pnc indexing/source code dev/api/visionet.api/visionet.api.common/models/
32	responsexmlpackage.cs	projects/pnq/source code dev/windows services/visionet.outboundintegration.service/visionet.outboundintegration.service/businessclasses/
33	screenhtmlhelper.cshtml	systems working/pnc indexing/source code dev/visionet.vlr/renderers/
34	screenhtmlhelperportal.cshtml	systems working/pnc indexing/source code dev/visionet.vlr/renderers/
35	searchsummarysheet.cs	projects/pnq/source code dev/windows services/visionet.outboundintegration.service/visionet.outboundintegration.service/businessclasses/
36	securitymanager.cs	projects/pnq/source code dev/visionet.vlr/visionnet.vlr.logging/
37	securitymanager.cs	systems working/pnc indexing/source code dev/visionet.vlr/visionnet.vlr.logging/
38	smtpemailer.vb	systems working/pnc indexing/source code dev/visionet.vlr/vlr_backend/visionet.email/common/
39	sqlspconstants.vb	systems working/pnc indexing/source code dev/visionet.vlr/vlr_backend/vlr.common/
40	symmetriccryptographer.cs	temp/pnc indexing/visionet.outboundintegration.service/vsiencompassconnect/dal/
41	utility.vb	systems working/pnc indexing/source code dev/visionet.vlr/vlr_backend/vlr.common/
42	xmlutil.cs	systems working/pnc indexing/source code dev/windows services/visionet.orderplacement.service/visionet.orderplacement.service/common/

Appendix C: Referenced Classpaths

Id	Path
1	document_splitting_service_exe.Document.Splitting.Service.AWSS3Call
2	document_splitting_service_exe.Document.Splitting.Service.DALHelper
3	document_splitting_service_exe.Document.Splitting.Service.Modules.AlfrescoSplitDocuments._3C_3Ec
4	document_splitting_service_exe.Document.Splitting.Service.Modules.SplitDocuments
5	document_splitting_service_exe.Document.Splitting.Service.Utility
6	kofax_omnipagecsdk_argtypes_dll.Kofax.OmniPageCSDK.ArgTypes.IOStreamCB
7	kofax_omnipagecsdk_objects_dll.Kofax.OmniPageCSDK.Objects.DataRule
8	visionet_vlr_common_dll.Visionet.VLR.Common.SecurityManager
9	visionet_vlr_common_dll.Visionet.VLR.Common.VLRConstants
10	visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.AutoMergeDocumentBL
11	visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.BL.AutoMergeForLenderLoanNumberBL
12	visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.BL.OCRFileOrderBL
13	visionet_vlr_ocrfileorder_backend_dll.Visionet.VLR.OCRFileOrder.Backend.FileOrder
14	visionet_vlr_web_presentation_dll.FileDownloadInMvc3.Models.FileDownloadResult
15	visionet_vlr_web_presentation_dll.Saml.Response
16	visionet_vlr_web_presentation_dll.Saml.Saml2DecryptResponse
17	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.AdminController
18	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.CommunicationPortalController
19	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.Dashboard.DashboardController
20	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.ExceptionManagementController
21	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.LoanReviewController
22	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.LoanTaskController
23	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.LoginController
24	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.LoginController._3CGetRefreshTokenInfo_3Ed__45
25	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.OCRDataMiningController
26	visionet_vlr_web_presentation_dll.Visionet.VLR.Web.Presentation.Controllers.VisiTrackController
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.Encryptor
33	vlr_classlibrary_dll.ImportUsersBLL
34	vlr_classlibrary_dll.VLR.ClassLibrary.CommonFunsBLL
35	vlr_classlibrary_dll.VLR.ClassLibrary.EmailManager
36	vlr_classlibrary_dll.VLR.ClassLibrary.Encryption.Asymmetric
37	vlr_classlibrary_dll.VLR.ClassLibrary.Encryption.Hash

Id	Path
38	vlr_classlibrary_dll.VLR.ClassLibrary.Encryption.Symmetric
39	vlr_classlibrary_dll.VLR.TapeCracking.FileSourceDAL
40	vlr_classlibrary_dll.VLR.TapeCracking.FileUploadHistoryBLL
41	vlr_classlibrary_dll.VLR.TapeCracking.cXLTemplateDAL
42	vlr_classlibrary_dll.cEncryptor
43	vlr_classlibrary_dll.cFTP
44	vlr_classlibrary_dll.cFTPService
45	vlr_common_dll.VLR.Common.CommonLib
46	vlr_common_dll.VLR.Common.Encryptor
47	vlr_common_dll.VLR.Common.Utility
48	vlr_common_dll.VLR.Common.cReportManager

Appendix D: Dynamic Flaw Inventory

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