MICRO FOCUS®

Fortify Security Report

2024-6-21

ASUS

Executive Summary

Issues Overview

On 2024-6-21, a source code review was performed over the xLua code base. 214 files, 15,247 LOC (Executable) were scanned and reviewed for defects that could lead to potential security vulnerabilities. A total of 2 reviewed findings were uncovered during the analysis.

Issues by Fortify Priority Order

High	2

Recommendations and Conclusions

The Issues Category section provides Fortify recommendations for addressing issues at a generic level. The recommendations for specific fixes can be extrapolated from those generic recommendations by the development group.

Project Summary

Code Base Summary

Code location: C:/Users/ASUS/Desktop/Gitrepo/xLua
Number of Files: 214
Lines of Code: 15247
Build Label: <No Build Label>

Scan Information

Scan time: 02:50
SCA Engine version: 20.1.1.0007
Machine Name: DESKTOP-MK5UPFE
Username running scan: ASUS

Results Certification

Results Certification Valid

Details:

Results Signature:

SCA Analysis Results has Valid signature

Rules Signature:

There were no custom rules used in this scan

Attack Surface

Attack Surface:
Command Line Arguments:
null.Main.Main
XLua.FilesSignature.Main
XLua.KeyPairsGen.Main
XLua.XLuaGenerate.Main
XLua.XLuaHotfixInject.Main

Filter Set Summary

Current Enabled Filter Set:
Quick View

Filter Set Details:

Folder Filters:
If [fortify priority order] contains critical Then set folder to Critical
If [fortify priority order] contains high Then set folder to High
If [fortify priority order] contains medium Then set folder to Medium
If [fortify priority order] contains low Then set folder to Low
Visibility Filters:

If impact is not in range [2.5, 5.0] Then hide issue
If likelihood is not in range (1.0, 5.0] Then hide issue

Audit Guide Summary

J2EE Bad Practices

Hide warnings about J2EE bad practices.
Depending on whether your application is a J2EE application, J2EE bad practice warnings may or may not apply. AuditGuide can hide J2EE bad practice warnings.
Enable if J2EE bad practice warnings do not apply to your application because it is not a J2EE application.

Filters:
If category contains j2ee Then hide issue
If category is race condition: static database connection Then hide issue

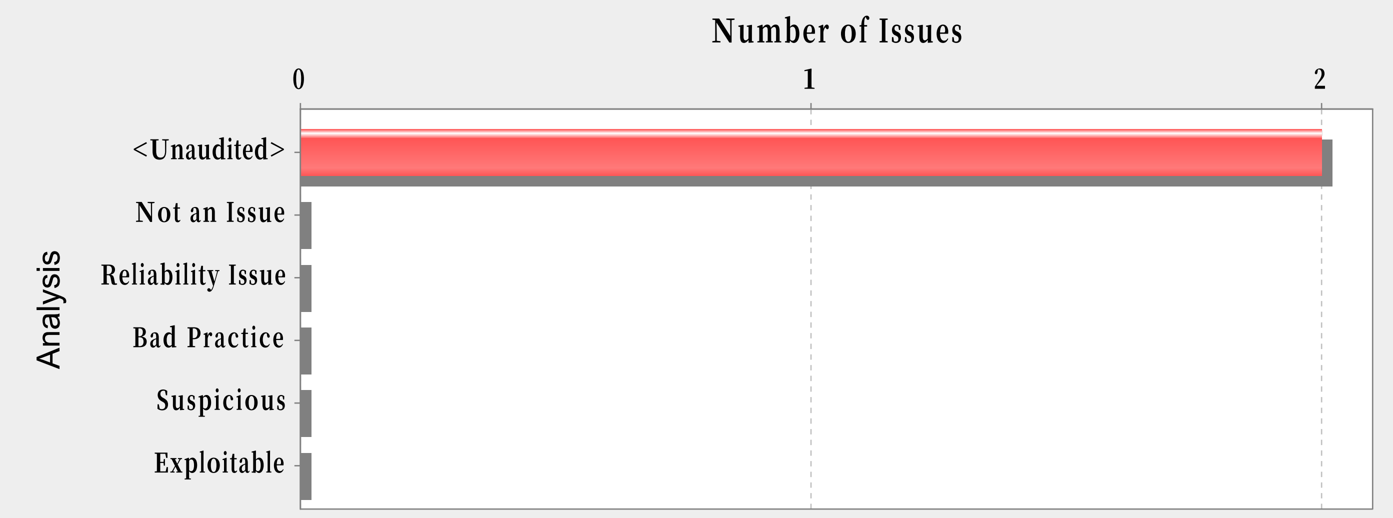
Results Outline

Overall number of results

The scan found 2 issues.

Vulnerability Examples by Category

Category: Key Management: Empty Encryption Key (2 Issues)



Abstract:

空加密密钥可能会削弱安全性，一旦出现安全问题将无法轻易修正。

Explanation:

使用空加密密钥绝非好方法，因为这样将会大幅减弱由良好的加密算法提供的保护，而且还会使解决这一问题变得极其困难。在问题代码投入使用之后，除非对软件进行修补，否则将无法更改空加密密钥。如果受空加密密钥保护的帐户遭受入侵，系统所有者将必须在安全性和可用性之间做出选择。

示例 1：以下代码使用空加密密钥执行 AES 加密：

```
...
var crypto = require('crypto');
var encryptionKey = "";
var algorithm = 'aes-256-ctr';
var cipher = crypto.createCipher(algorithm, encryptionKey);
...
```

不仅任何可以访问此代码的人可以确定它使用空加密密钥，而且任何具有最基本破解技术的人都更有可能成功解密所有加密数据。在应用程序发布之后，必须对软件进行修补才能更改空加密密钥。雇员可以利用手中掌握的信息访问权限入侵系统。即使攻击者只能访问应用程序的可执行文件，他们也可以提取使用了空加密密钥的证据。

Recommendations:

加密密钥不能为空，而应对加密密钥加以模糊化，并在外部资源文件中进行管理。如果在系统中采用明文的形式存储加密密钥（空或非空），任何有足够权限的人即可读取加密密钥，还可能误用这些加密密钥。

vue.js, line 8844 (Key Management: Empty Encryption Key)

Fortify Priority:	High	**Folder**	High
Kingdom:	Security Features		
Abstract:	空加密密钥可能会削弱安全性，一旦出现安全问题将无法轻易修正。		
Sink:	vue.js:8844 VariableAccess: key()		

```
8842      return genIf(el)
8843    } else if (el.staticInFor) {
8844      var key = '';
8845      var parent = el.parent;
8846      while (parent) {
```

Issue Count by Category

Issues by Category	
Key Management: Empty Encryption Key	2

Issue Breakdown by Analysis

Issues by Analysis