



Fortify Security Report

2024-6-21

ASUS

Executive Summary

Issues Overview

On 2024-6-21, a source code review was performed over the zfs code base. 29 files, 5,655 LOC (Executable) were scanned and reviewed for defects that could lead to potential security vulnerabilities. A total of 7 reviewed findings were uncovered during the analysis.

Issues by Fortify Priority Order

Critical	4
High	3

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/zfs

Number of Files: 29

Lines of Code: 5655

Build Label: <No Build Label>

Scan Information

Scan time: 01:21

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:

File System:

os.null.open

Stream:

os.null.read

System Information:

null.null.null

null.null.null

null.null.globals

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

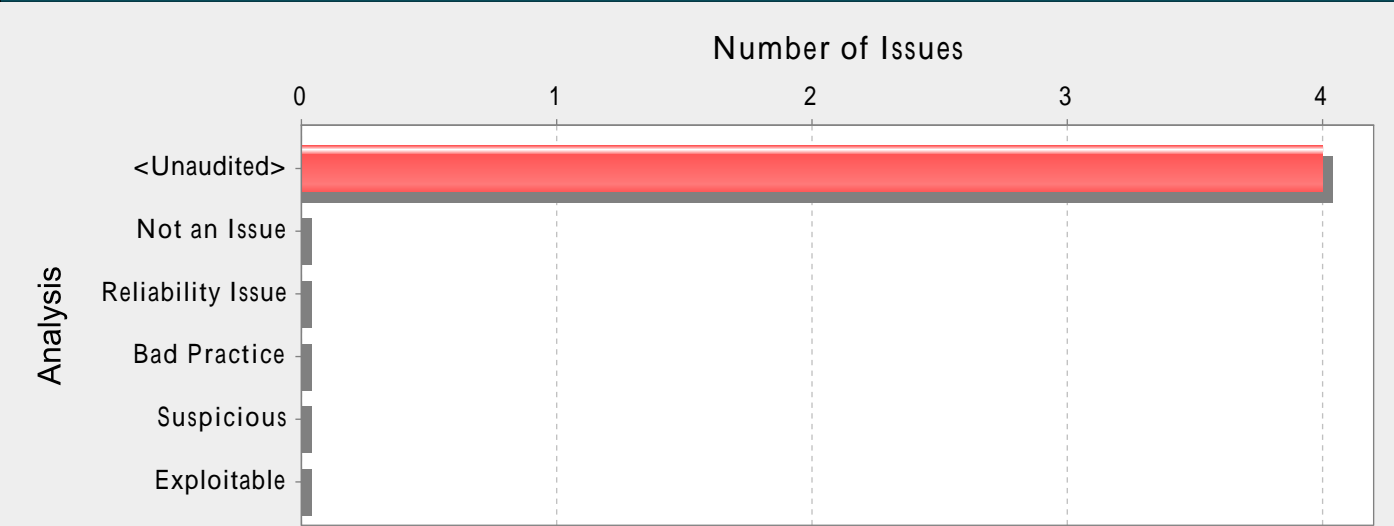
Results Outline

Overall number of results

The scan found 7 issues.

Vulnerability Examples by Category

Category: Key Management: Empty Encryption Key (4 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。这不仅是因为使用空加密密钥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：以下代码会将加密密钥变量初始化为空字符串。

```
...
from Crypto.Ciphers import AES
cipher = AES.new("", AES.MODE_CFB, iv)
msg = iv + cipher.encrypt(b'Attack at dawn')
...
```

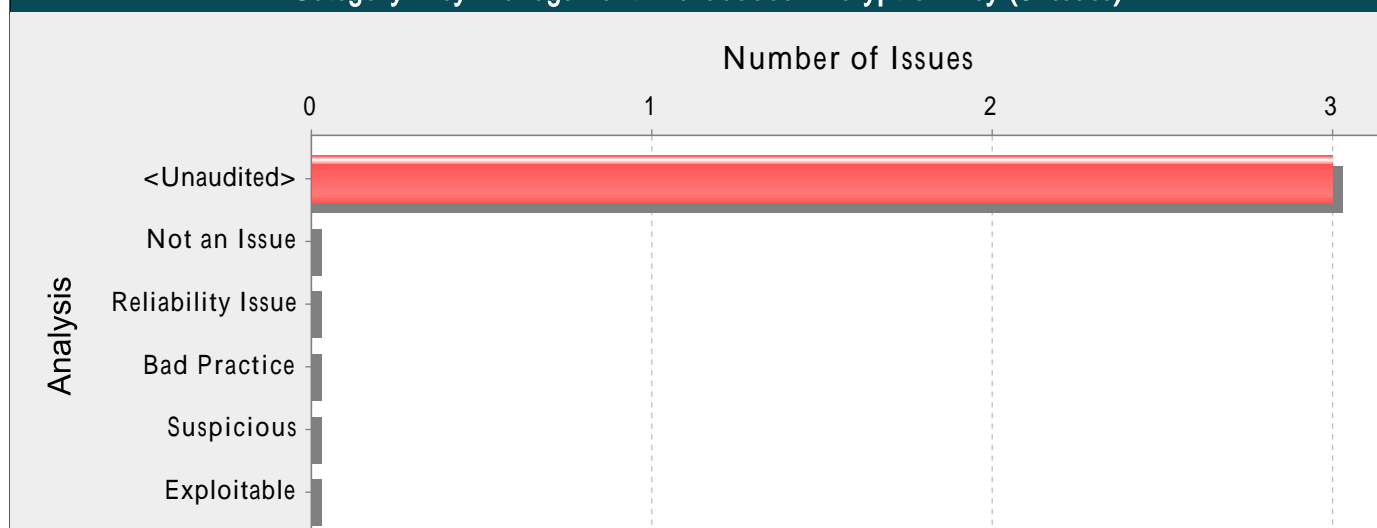
不仅任何可以访问此代码的人可以确定它使用的是空加密密钥，而且任何掌握最基本破解技术的人都更有可能成功解密所有加密数据。一旦程序发布，要更改空加密密钥，就必须进行软件修补。雇员可以利用手中掌握的信息访问权限入侵系统。即使攻击者只能访问应用程序的可执行文件，他们也可以提取使用了空加密密钥的证据。

Recommendations:

加密密钥绝不能为空。通常情况下，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。

_libzfs_core.py, line 116 (Key Management: Empty Encryption Key)			
Fortify Priority:	Critical	Folder	Critical
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:	_libzfs_core.py:116 VariableAccess: key()		
114	props = {}		
115	if key is None:		
116	key = b""		
117	else:		
118	key = bytes(key)		

Category: Key Management: Hardcoded Encryption Key (3 Issues)

**Abstract:**

Hardcoded 加密密钥可能会削弱系统安全性，一旦出现安全问题将无法轻易修正。

Explanation:

使用硬编码方式处理加密密钥绝非好方法。这不仅是因为所有项目开发人员都可以使用通过硬编码方式处理的加密密钥，而且还会使解决这一问题变得极其困难。在代码投入使用之后，必须对软件进行修补才能更改加密密钥。如果受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：下列代码使用 hardcoded 加密密钥来加密信息：

```
...
from Crypto.Ciphers import AES
encryption_key = b'_hardcoded__key_'
cipher = AES.new(encryption_key, AES.MODE_CFB, iv)
msg = iv + cipher.encrypt(b'Attack at dawn')
...
```

此代码将成功运行，但任何有权访问此代码的人都可以获得加密密钥。一旦程序发布，除非修补该程序，否则可能无法更改硬编码的加密密钥 _hardcoded__key_。心怀不轨的雇员可以利用其对此信息的访问权限来破坏系统加密的数据。

Recommendations:

绝不能对加密密钥进行硬编码。通常情况下，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，任何有足够权限的人即可读取加密密钥，还可能误用这些密码。

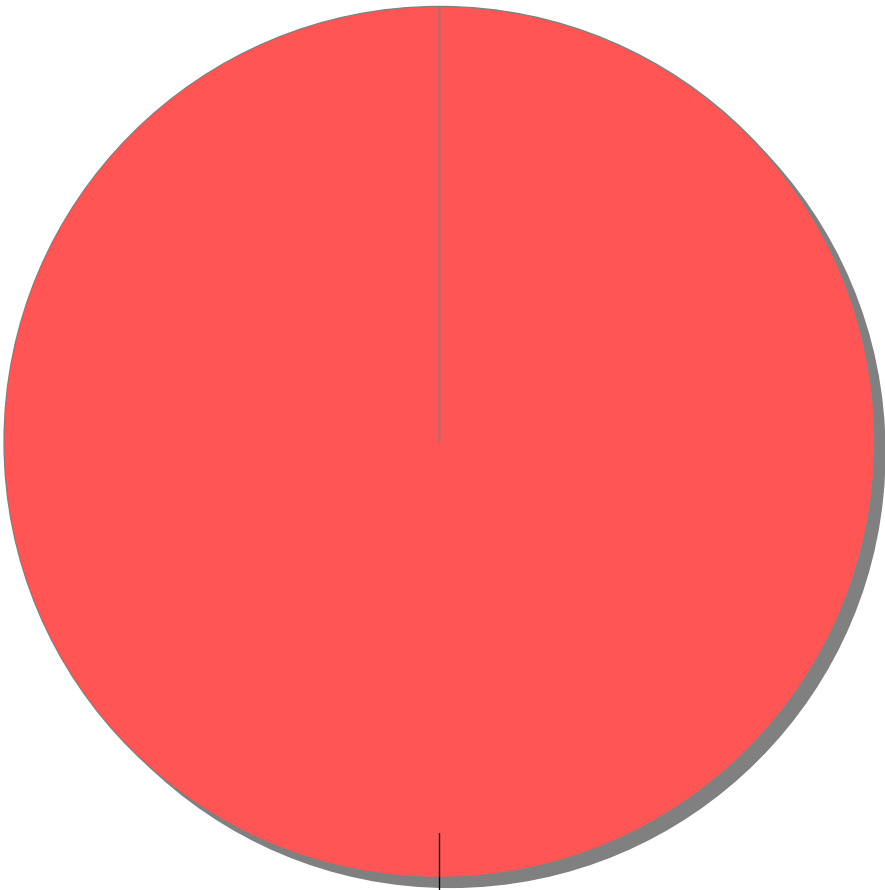
exceptions.py, line 517 (Key Management: Hardcoded Encryption Key)

Fortify Priority:	High	Folder	High
Kingdom:	Security Features		
Abstract:	Hardcoded 加密密钥可能会削弱系统安全性，一旦出现安全问题将无法轻易修正。		
Sink:	exceptions.py:517 VariableAccess: EncryptionKeyNotLoaded()		
515	class EncryptionKeyNotLoaded(ZFSError):		
516	errno = errno.EACCES		
517	message = "Encryption key is not currently loaded"		


Issue Count by Category	
Issues by Category	
Key Management: Empty Encryption Key	4
Key Management: Hardcoded Encryption Key	3

Issue Breakdown by Analysis

Issues by Analysis



<none>: (7, 100%)

 <none>