Fortify Standalone Report Generator

Developer Workbook

akka-distributed-data

Table of Contents

© Copyright [2008-2022] Micro Focus or one of its affiliates.

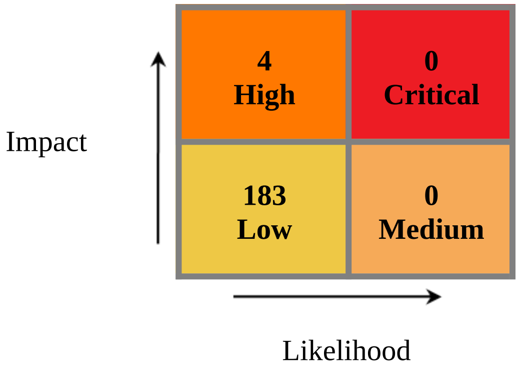
Executive Summary

This workbook is intended to provide all necessary details and information for a developer to understand and remediate the different issues discovered during the akka-distributed-data project audit. The information contained in this workbook is targeted at project managers and developers.

This section provides an overview of the issues uncovered during analysis.

Project Name:	akka-distributed-data
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
WebInspect Agent:	Results Not Present
Other:	Results Not Present

Issues by Priority



Top Ten Critical Categories

This project does not contain any critical issues

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Project Description

This section provides an overview of the Fortify scan engines used for this project, as well as the project meta-information.

SCA

Date of Last Analysis:	Jun 16, 2022, 11:29 AM	**Engine Version:**	21.1.1.0009
Host Name:	Jacks-Work-MBP.local	**Certification:**	VALID
Number of Files:	44	**Lines of Code:**	6,771

Rulepack Name	Rulepack Version
Fortify Secure Coding Rules, Extended, Java	2022.1.0.0007
Fortify Secure Coding Rules, Core, Scala	2022.1.0.0007
Fortify Secure Coding Rules, Extended, JSP	2022.1.0.0007
Fortify Secure Coding Rules, Core, Android	2022.1.0.0007
Fortify Secure Coding Rules, Extended, Content	2022.1.0.0007
Fortify Secure Coding Rules, Extended, Configuration	2022.1.0.0007
Fortify Secure Coding Rules, Core, Annotations	2022.1.0.0007
Fortify Secure Coding Rules, Community, Cloud	2022.1.0.0007
Fortify Secure Coding Rules, Core, Universal	2022.1.0.0007
Fortify Secure Coding Rules, Core, Java	2022.1.0.0007
Fortify Secure Coding Rules, Community, Universal	2022.1.0.0007

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Issue Breakdown by Fortify Categories

The following table depicts a summary of all issues grouped vertically by Fortify Category. For each category, the total number of issues is shown by Fortify Priority Order, including information about the number of audited issues.

Category	Fortify Priority (audited/total)				Total Issues
	Critical	High	Medium	Low	
Code Correctness: Constructor Invokes Overridable Function	0	0	0	0 / 118	0 / 118
Code Correctness: Erroneous String Compare	0	0	0	0 / 4	0 / 4
Code Correctness: Non-Static Inner Class Implements Serializable	0	0	0	0 / 45	0 / 45
Dead Code: Expression is Always false	0	0	0	0 / 11	0 / 11
Insecure Randomness	0	0 / 4	0	0	0 / 4
J2EE Bad Practices: Leftover Debug Code	0	0	0	0 / 1	0 / 1
J2EE Bad Practices: Threads	0	0	0	0 / 1	0 / 1
Poor Style: Value Never Read	0	0	0	0 / 1	0 / 1
Unchecked Return Value	0	0	0	0 / 1	0 / 1
Weak Cryptographic Hash	0	0	0	0 / 1	0 / 1

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Results Outline

Code Correctness: Constructor Invokes Overridable Function (118 issues)

Abstract

A constructor of the class calls a function that can be overridden.

Explanation

When a constructor calls an overridable function, it may allow an attacker to access the `this` reference prior to the object being fully initialized, which can in turn lead to a vulnerability. **Example 1:** The following calls a method that can be overridden.

```
...
class User {
  private String username;
  private boolean valid;
  public User(String username, String password){
    this.username = username;
    this.valid = validateUser(username, password);
  }
  public boolean validateUser(String username, String password){
    //validate user is real and can authenticate
    ...
  }
  public final boolean isValid(){
    return valid;
  }
}
```

Since the function `validateUser` and the class are not `final`, it means that they can be overridden, and then initializing a variable to the subclass that overrides this function would allow bypassing of the `validateUser` functionality. For example:

```
...
class Attacker extends User{
  public Attacker(String username, String password){
    super(username, password);
  }
  public boolean validateUser(String username, String password){
    return true;
  }
}
...
class MainClass{
  public static void main(String[] args){
    User hacker = new Attacker("Evil", "Hacker");
    if (hacker.isValid()){
      System.out.println("Attack successful!");
    }else{
      System.out.println("Attack failed");
    }
  }
}
```

The code in `Example 1` prints "Attack successful!", since the `Attacker` class overrides the `validateUser()` function that is called from the constructor of the superclass `User`, and Java will first look in the subclass for functions called from the constructor.

© Copyright [2008-2022] Micro Focus or one of its affiliates.

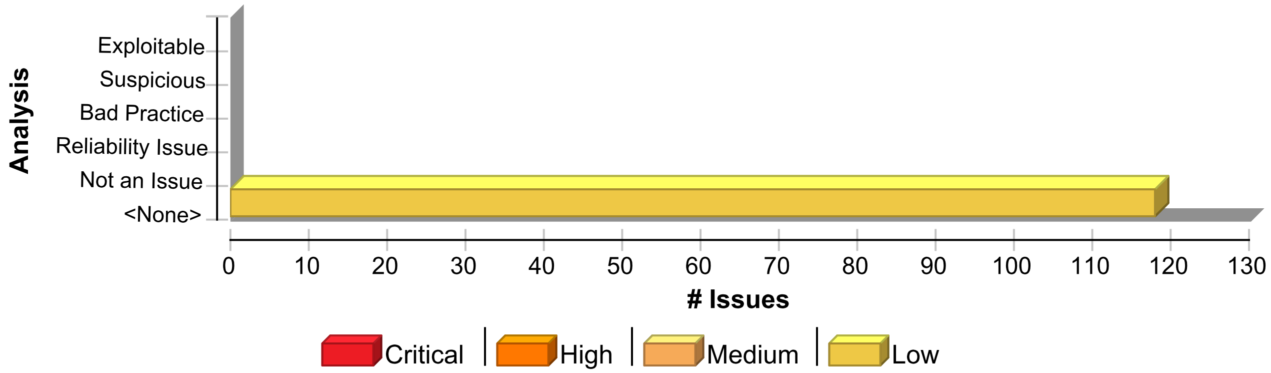
Recommendation

Constructors should not call functions that can be overridden, either by specifying them as `final`, or specifying the class as `final`. Alternatively if this code is only ever needed in the constructor, the `private` access specifier can be used, or the logic could be placed directly into the constructor of the superclass. **Example 2:** The following makes the class `final` to prevent the function from being overridden elsewhere.

```
  ...
  final class User {
    private String username;
    private boolean valid;
    public User(String username, String password){
      this.username = username;
      this.valid = validateUser(username, password);
    }
    private boolean validateUser(String username, String password){
      //validate user is real and can authenticate
      ...
    }
    public final boolean isValid(){
      return valid;
    }
  }
```

This example specifies the class as `final`, so that it cannot be subclassed, and changes the `validateUser()` function to `private`, since it is not needed elsewhere in this application. This is programming defensively, since at a later date it may be decided that the `User` class needs to be subclassed, which would result in this vulnerability reappearing if the `validateUser()` function was not set to `private`.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Constructor Invokes Overridable Function	118	0	0	118
Total	**118**	**0**	**0**	**118**

Code Correctness: Constructor Invokes Overridable Function **Low**

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 151 (Code Correctness: Constructor Invokes Overridable Function) **Low**

Issue Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 151 (Code Correctness: Constructor Invokes Overridable Function) — Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeB
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:151
Taint Flags:

```
148  val nodeC = UniqueAddress(Address(protocol, "Sys", "c", 2552), 17L)
149  val nodeD = UniqueAddress(Address(protocol, "Sys", "d", 2552), 17L)
150  // 4 replicas + the local => 5
151  val nodes = Vector(nodeA, nodeB, nodeC, nodeD)
152
153  val data = GSet.empty + "A" + "B"
154  val timeout = 3.seconds.dilated
```

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 151 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeA
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:151
Taint Flags:

```
148  val nodeC = UniqueAddress(Address(protocol, "Sys", "c", 2552), 17L)
149  val nodeD = UniqueAddress(Address(protocol, "Sys", "d", 2552), 17L)
150  // 4 replicas + the local => 5
151  val nodes = Vector(nodeA, nodeB, nodeC, nodeD)
152
153  val data = GSet.empty + "A" + "B"
154  val timeout = 3.seconds.dilated
```

test/scala/akka/cluster/ddata/PruningStateSpec.scala, line 17 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/PruningStateSpec.scala, line 17 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: node1
Enclosing Method: PruningStateSpec()
File: test/scala/akka/cluster/ddata/PruningStateSpec.scala:17
Taint Flags:

```
14  import PruningState._
15
16  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
17  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
18  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
19  val node4 = UniqueAddress(node1.address.copy(port = Some(2554)), 4L)
20
```

test/scala/akka/cluster/ddata/PruningStateSpec.scala, line 18 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: PruningStateSpec()
File: test/scala/akka/cluster/ddata/PruningStateSpec.scala:18
Taint Flags:

```
15
16  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
17  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
18  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
19  val node4 = UniqueAddress(node1.address.copy(port = Some(2554)), 4L)
20
21  "Pruning state" must {
```

test/scala/akka/cluster/ddata/PruningStateSpec.scala, line 19 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: PruningStateSpec()

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/PruningStateSpec.scala, line 19 (Code Correctness: Constructor Invokes Overridable Function)

Low

File: test/scala/akka/cluster/ddata/PruningStateSpec.scala:19
Taint Flags:

```
16  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
17  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
18  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
19  val node4 = UniqueAddress(node1.address.copy(port = Some(2554)), 4L)
20
21  "Pruning state" must {
22
```

test/scala/akka/cluster/ddata/PNCounterMapSpec.scala, line 17 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: PNCounterMapSpec()
File: test/scala/akka/cluster/ddata/PNCounterMapSpec.scala:17
Taint Flags:

```
14 class PNCounterMapSpec extends AnyWordSpec with Matchers {
15
16  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
17  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
18
19  "A PNCounterMap" must {
20
```

main/scala/akka/cluster/ddata/Replicator.scala, line 1365 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: selfUniqueAddress
Enclosing Method: Replicator()
File: main/scala/akka/cluster/ddata/Replicator.scala:1365
Taint Flags:

```
1362  val cluster = Cluster(context.system)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 1365 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
1363  val selfAddress = cluster.selfAddress
1364  val selfUniqueAddress = cluster.selfUniqueAddress
1365  val selfFromSystemUid = Some(selfUniqueAddress.longUid)
1366
1367  require(!cluster.isTerminated, "Cluster node must not be terminated")
1368  require(
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeA
Enclosing Method: ORSetSpec()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:23
Taint Flags:

```
20  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
21
22  val nodeA = UniqueAddress(Address("akka", "Sys", "a", 2552), 1L)
23  val nodeB = UniqueAddress(nodeA.address.copy(host = Some("b")), 2L)
24  val nodeC = UniqueAddress(nodeA.address.copy(host = Some("c")), 3L)
25  val nodeD = UniqueAddress(nodeA.address.copy(host = Some("d")), 4L)
26  val nodeE = UniqueAddress(nodeA.address.copy(host = Some("e")), 5L)
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 24 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeA
Enclosing Method: ORSetSpec()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:24
Taint Flags:

```
21
22  val nodeA = UniqueAddress(Address("akka", "Sys", "a", 2552), 1L)
23  val nodeB = UniqueAddress(nodeA.address.copy(host = Some("b")), 2L)
24  val nodeC = UniqueAddress(nodeA.address.copy(host = Some("c")), 3L)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 24 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
25  val nodeD = UniqueAddress(nodeA.address.copy(host = Some("d")), 4L)
26  val nodeE = UniqueAddress(nodeA.address.copy(host = Some("e")), 5L)
27  val nodeF = UniqueAddress(nodeA.address.copy(host = Some("f")), 6L)
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 25 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeA
Enclosing Method: ORSetSpec()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:25
Taint Flags:

```
22  val nodeA = UniqueAddress(Address("akka", "Sys", "a", 2552), 1L)
23  val nodeB = UniqueAddress(nodeA.address.copy(host = Some("b")), 2L)
24  val nodeC = UniqueAddress(nodeA.address.copy(host = Some("c")), 3L)
25  val nodeD = UniqueAddress(nodeA.address.copy(host = Some("d")), 4L)
26  val nodeE = UniqueAddress(nodeA.address.copy(host = Some("e")), 5L)
27  val nodeF = UniqueAddress(nodeA.address.copy(host = Some("f")), 6L)
28  val nodeG = UniqueAddress(nodeA.address.copy(host = Some("g")), 7L)
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 26 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeA
Enclosing Method: ORSetSpec()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:26
Taint Flags:

```
23  val nodeB = UniqueAddress(nodeA.address.copy(host = Some("b")), 2L)
24  val nodeC = UniqueAddress(nodeA.address.copy(host = Some("c")), 3L)
25  val nodeD = UniqueAddress(nodeA.address.copy(host = Some("d")), 4L)
26  val nodeE = UniqueAddress(nodeA.address.copy(host = Some("e")), 5L)
27  val nodeF = UniqueAddress(nodeA.address.copy(host = Some("f")), 6L)
28  val nodeG = UniqueAddress(nodeA.address.copy(host = Some("g")), 7L)
29  val nodeH = UniqueAddress(nodeA.address.copy(host = Some("h")), 8L)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 27 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeA
Enclosing Method: ORSetSpec()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:27
Taint Flags:

```
24  val nodeC = UniqueAddress(nodeA.address.copy(host = Some("c")), 3L)
25  val nodeD = UniqueAddress(nodeA.address.copy(host = Some("d")), 4L)
26  val nodeE = UniqueAddress(nodeA.address.copy(host = Some("e")), 5L)
27  val nodeF = UniqueAddress(nodeA.address.copy(host = Some("f")), 6L)
28  val nodeG = UniqueAddress(nodeA.address.copy(host = Some("g")), 7L)
29  val nodeH = UniqueAddress(nodeA.address.copy(host = Some("h")), 8L)
30
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 28 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeA
Enclosing Method: ORSetSpec()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:28
Taint Flags:

```
25  val nodeD = UniqueAddress(nodeA.address.copy(host = Some("d")), 4L)
26  val nodeE = UniqueAddress(nodeA.address.copy(host = Some("e")), 5L)
27  val nodeF = UniqueAddress(nodeA.address.copy(host = Some("f")), 6L)
28  val nodeG = UniqueAddress(nodeA.address.copy(host = Some("g")), 7L)
29  val nodeH = UniqueAddress(nodeA.address.copy(host = Some("h")), 8L)
30
31  val user1 = """{"username":"john","password":"coltrane"}"""
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 29 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 29 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: nodeA
Enclosing Method: ORSetSpec()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:29
Taint Flags:

```
26  val nodeE = UniqueAddress(nodeA.address.copy(host = Some("e")), 5L)
27  val nodeF = UniqueAddress(nodeA.address.copy(host = Some("f")), 6L)
28  val nodeG = UniqueAddress(nodeA.address.copy(host = Some("g")), 7L)
29  val nodeH = UniqueAddress(nodeA.address.copy(host = Some("h")), 8L)
30
31  val user1 = """{"username":"john","password":"coltrane"}"""
32  val user2 = """{"username":"sonny","password":"rollins"}"""
```

main/scala/akka/cluster/ddata/GCounter.scala, line 14 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: $default$1
Enclosing Method: GCounter()
File: main/scala/akka/cluster/ddata/GCounter.scala:14
Taint Flags:

```
11 import akka.cluster.UniqueAddress
12
13 object GCounter {
14  val empty: GCounter = new GCounter
15  def apply(): GCounter = empty
16
17  /**
```

main/scala/akka/cluster/ddata/ORSet.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: $default$3
Enclosing Method: ORSet()

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/ORSet.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: main/scala/akka/cluster/ddata/ORSet.scala:16
Taint Flags:

```
13 import akka.util.{ unused, HashCode }
14
15 object ORSet {
16  private val _empty: ORSet[Any] = new ORSet(Map.empty, VersionVector.empty)
17  def empty[A]: ORSet[A] = _empty.asInstanceOf[ORSet[A]]
18  def apply(): ORSet[Any] = _empty
19
```

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 151 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeC
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:151
Taint Flags:

```
148  val nodeC = UniqueAddress(Address(protocol, "Sys", "c", 2552), 17L)
149  val nodeD = UniqueAddress(Address(protocol, "Sys", "d", 2552), 17L)
150  // 4 replicas + the local => 5
151  val nodes = Vector(nodeA, nodeB, nodeC, nodeD)
152
153  val data = GSet.empty + "A" + "B"
154  val timeout = 3.seconds.dilated
```

test/scala/akka/cluster/ddata/LWWRegisterSpec.scala, line 18 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: LWWRegisterSpec()
File: test/scala/akka/cluster/ddata/LWWRegisterSpec.scala:18
Taint Flags:

```
15  import LWWRegister.defaultClock
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/LWWRegisterSpec.scala, line 18 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
16
17  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
18  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
19
20  "A LWWRegister" must {
21  "use latest of successive assignments" in {
```

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 151 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeD
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:151
Taint Flags:

```
148  val nodeC = UniqueAddress(Address(protocol, "Sys", "c", 2552), 17L)
149  val nodeD = UniqueAddress(Address(protocol, "Sys", "d", 2552), 17L)
150  // 4 replicas + the local => 5
151  val nodes = Vector(nodeA, nodeB, nodeC, nodeD)
152
153  val data = GSet.empty + "A" + "B"
154  val timeout = 3.seconds.dilated
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 19 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: ORSetSpec()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:19
Taint Flags:

```
16 class ORSetSpec extends AnyWordSpec with Matchers {
17
18  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
19  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 19 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
20 val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
21
22 val nodeA = UniqueAddress(Address("akka", "Sys", "a", 2552), 1L)
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 20 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: ORSetSpec()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:20
Taint Flags:

```
17
18 val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
19 val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
20 val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
21
22 val nodeA = UniqueAddress(Address("akka", "Sys", "a", 2552), 1L)
23 val nodeB = UniqueAddress(nodeA.address.copy(host = Some("b")), 2L)
```

main/scala/akka/cluster/ddata/Flag.scala, line 17 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: empty
Enclosing Method: Flag()
File: main/scala/akka/cluster/ddata/Flag.scala:17
Taint Flags:

```
14 /**
15 * `Flag` that is initialized to `false`.
16 */
17 val Disabled: Flag = empty
18
19 /**
20 * `Flag` that is initialized to `true`.
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/ORMapSpec.scala, line 18 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: ORMapSpec()
File: test/scala/akka/cluster/ddata/ORMapSpec.scala:18
Taint Flags:

```
15 class ORMapSpec extends AnyWordSpec with Matchers {
16
17  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
18  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
19
20  "A ORMap" must {
21
```

main/scala/akka/cluster/ddata/DurableStore.scala, line 121 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: serializer
Enclosing Method: LmdbDurableStore()
File: main/scala/akka/cluster/ddata/DurableStore.scala:121
Taint Flags:

```
118
119  val serialization = SerializationExtension(context.system)
120  val serializer = serialization.serializerFor(classOf[DurableDataEnvelope]).asInstanceOf[SerializerWithStringManifest]
121  val manifest = serializer.manifest(new DurableDataEnvelope(Replicator.Internal.DeletedData))
122
123  val writeBehindInterval = config.getString("lmdb.write-behind-interval").toLowerCase match {
124  case "off" => Duration.Zero
```

main/scala/akka/cluster/ddata/EstimatedSize.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/EstimatedSize.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: Address
Enclosing Method: EstimatedSize()
File: main/scala/akka/cluster/ddata/EstimatedSize.scala:16
Taint Flags:

```
13 @InternalApi private[akka] object EstimatedSize {
14  val LongValue = 8
15  val Address = 50
16  val UniqueAddress = Address + LongValue
17 }
18
19 undefined
```

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 146 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: protocol
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:146
Taint Flags:

```
143  if (RARP(system).provider.remoteSettings.Artery.Enabled) "akka"
144  else "akka.tcp"
145
146  val nodeA = UniqueAddress(Address(protocol, "Sys", "a", 2552), 17L)
147  val nodeB = UniqueAddress(Address(protocol, "Sys", "b", 2552), 17L)
148  val nodeC = UniqueAddress(Address(protocol, "Sys", "c", 2552), 17L)
149  val nodeD = UniqueAddress(Address(protocol, "Sys", "d", 2552), 17L)
```

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 147 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: protocol
Enclosing Method: WriteAggregatorSpec()

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 147 (Code Correctness: Constructor Invokes Overridable Function)

Low

File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:147
Taint Flags:

```
144  else "akka.tcp"
145
146  val nodeA = UniqueAddress(Address(protocol, "Sys", "a", 2552), 17L)
147  val nodeB = UniqueAddress(Address(protocol, "Sys", "b", 2552), 17L)
148  val nodeC = UniqueAddress(Address(protocol, "Sys", "c", 2552), 17L)
149  val nodeD = UniqueAddress(Address(protocol, "Sys", "d", 2552), 17L)
150  // 4 replicas + the local => 5
```

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 148 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: protocol
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:148
Taint Flags:

```
145
146  val nodeA = UniqueAddress(Address(protocol, "Sys", "a", 2552), 17L)
147  val nodeB = UniqueAddress(Address(protocol, "Sys", "b", 2552), 17L)
148  val nodeC = UniqueAddress(Address(protocol, "Sys", "c", 2552), 17L)
149  val nodeD = UniqueAddress(Address(protocol, "Sys", "d", 2552), 17L)
150  // 4 replicas + the local => 5
151  val nodes = Vector(nodeA, nodeB, nodeC, nodeD)
```

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 149 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: protocol
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:149
Taint Flags:

```
146  val nodeA = UniqueAddress(Address(protocol, "Sys", "a", 2552), 17L)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 149 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
147  val nodeB = UniqueAddress(Address(protocol, "Sys", "b", 2552), 17L)
148  val nodeC = UniqueAddress(Address(protocol, "Sys", "c", 2552), 17L)
149  val nodeD = UniqueAddress(Address(protocol, "Sys", "d", 2552), 17L)
150  // 4 replicas + the local => 5
151  val nodes = Vector(nodeA, nodeB, nodeC, nodeD)
152
```

main/scala/akka/cluster/ddata/GCounter.scala, line 14 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: $default$2
Enclosing Method: GCounter()
File: main/scala/akka/cluster/ddata/GCounter.scala:14
Taint Flags:

```
11 import akka.cluster.UniqueAddress
12
13 object GCounter {
14  val empty: GCounter = new GCounter
15  def apply(): GCounter = empty
16
17  /**
```

test/scala/akka/cluster/ddata/ORMultiMapSpec.scala, line 17 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: ORMultiMapSpec()
File: test/scala/akka/cluster/ddata/ORMultiMapSpec.scala:17
Taint Flags:

```
14 class ORMultiMapSpec extends AnyWordSpec with Matchers {
15
16  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
17  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/ORMultiMapSpec.scala, line 17 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
18
19  "A ORMultiMap" must {
20
```

main/scala/akka/cluster/ddata/Replicator.scala, line 1363 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: Replicator()
File: main/scala/akka/cluster/ddata/Replicator.scala:1363
Taint Flags:

```
1360  import settings._
1361
1362  val cluster = Cluster(context.system)
1363  val selfAddress = cluster.selfAddress
1364  val selfUniqueAddress = cluster.selfUniqueAddress
1365  val selfFromSystemUid = Some(selfUniqueAddress.longUid)
1366
```

main/scala/akka/cluster/ddata/Replicator.scala, line 1364 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: Replicator()
File: main/scala/akka/cluster/ddata/Replicator.scala:1364
Taint Flags:

```
1361
1362  val cluster = Cluster(context.system)
1363  val selfAddress = cluster.selfAddress
1364  val selfUniqueAddress = cluster.selfUniqueAddress
1365  val selfFromSystemUid = Some(selfUniqueAddress.longUid)
1366
1367  require(!cluster.isTerminated, "Cluster node must not be terminated")
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 1367 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: Replicator()
File: main/scala/akka/cluster/ddata/Replicator.scala:1367
Taint Flags:

```
1364 val selfUniqueAddress = cluster.selfUniqueAddress
1365 val selfFromSystemUid = Some(selfUniqueAddress.longUid)
1366
1367 require(!cluster.isTerminated, "Cluster node must not be terminated")
1368 require(
1369 roles.subsetOf(cluster.selfRoles),
1370 s"This cluster member [${selfAddress}] doesn't have all the roles [${roles.mkString(", ")}]")
```

main/scala/akka/cluster/ddata/Replicator.scala, line 1368 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: Replicator()
File: main/scala/akka/cluster/ddata/Replicator.scala:1368
Taint Flags:

```
1365 val selfFromSystemUid = Some(selfUniqueAddress.longUid)
1366
1367 require(!cluster.isTerminated, "Cluster node must not be terminated")
1368 require(
1369 roles.subsetOf(cluster.selfRoles),
1370 s"This cluster member [${selfAddress}] doesn't have all the roles [${roles.mkString(", ")}]")
1371
```

main/scala/akka/cluster/ddata/DurableStore.scala, line 120 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/DurableStore.scala, line 120 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: serialization
Enclosing Method: LmdbDurableStore()
File: main/scala/akka/cluster/ddata/DurableStore.scala:120
Taint Flags:

```
117  import LmdbDurableStore.WriteBehind
118
119  val serialization = SerializationExtension(context.system)
120  val serializer = serialization.serializerFor(classOf[DurableDataEnvelope]).asInstanceOf[SerializerWithStringManifest]
121  val manifest = serializer.manifest(new DurableDataEnvelope(Replicator.Internal.DeletedData))
122
123  val writeBehindInterval = config.getString("lmdb.write-behind-interval").toLowerCase match {
```

test/scala/akka/cluster/ddata/LWWMapSpec.scala, line 18 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: LWWMapSpec()
File: test/scala/akka/cluster/ddata/LWWMapSpec.scala:18
Taint Flags:

```
15  import LWWRegister.defaultClock
16
17  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
18  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
19
20  "A LWWMap" must {
21
```

main/scala/akka/cluster/ddata/EstimatedSize.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: LongValue
Enclosing Method: EstimatedSize()

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/EstimatedSize.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function)

Low

File: main/scala/akka/cluster/ddata/EstimatedSize.scala:16
Taint Flags:

```
13 @InternalApi private[akka] object EstimatedSize {
14  val LongValue = 8
15  val Address = 50
16  val UniqueAddress = Address + LongValue
17 }
18
19 undefined
```

main/scala/akka/cluster/ddata/Replicator.scala, line 1405 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: hasDurableKeys
Enclosing Method: Replicator()
File: main/scala/akka/cluster/ddata/Replicator.scala:1405
Taint Flags:

```
1402  val durable = settings.durableKeys.filterNot(_.endsWith("*"))
1403  val durableWildcards = settings.durableKeys.collect { case k if k.endsWith("*") => k.dropRight(1) }
1404  val durableStore: ActorRef =
1405  if (hasDurableKeys) {
1406  val props = settings.durableStoreProps match {
1407  case Right(p) => p
1408  case Left((s, c)) =>
```

main/scala/akka/cluster/ddata/DistributedData.scala, line 47 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: settings
Enclosing Method: DistributedData()
File: main/scala/akka/cluster/ddata/DistributedData.scala:47
Taint Flags:

```
44  if (Cluster(system).isTerminated)
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/DistributedData.scala, line 47 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
45 log.warning("Replicator points to dead letters, because Cluster is terminated.")
46 else
47 log.warning(
48 "Replicator points to dead letters. Make sure the cluster node has the proper role. " +
49 "Node has roles [{}], Distributed Data is configured for roles [{}].",
50 Cluster(system).selfRoles.mkString(","),
```

main/scala/akka/cluster/ddata/DistributedData.scala, line 54 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: settings
Enclosing Method: DistributedData()
File: main/scala/akka/cluster/ddata/DistributedData.scala:54
Taint Flags:

```
51 settings.roles.mkString(","): Any)
52 system.deadLetters
53 } else {
54 system.systemActorOf(Replicator.props(settings), ReplicatorSettings.name(system, None))
55 }
56
57 /**
```

test/scala/akka/cluster/ddata/GCounterSpec.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: GCounterSpec()
File: test/scala/akka/cluster/ddata/GCounterSpec.scala:16
Taint Flags:

```
13
14 class GCounterSpec extends AnyWordSpec with Matchers {
15 val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
16 val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/GCounterSpec.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function)

Low

```
17  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
18
19  "A GCounter" must {
```

test/scala/akka/cluster/ddata/GCounterSpec.scala, line 17 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: GCounterSpec()
File: test/scala/akka/cluster/ddata/GCounterSpec.scala:17
Taint Flags:

```
14 class GCounterSpec extends AnyWordSpec with Matchers {
15  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
16  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
17  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
18
19  "A GCounter" must {
20
```

test/scala/akka/cluster/ddata/DataEnvelopeSpec.scala, line 18 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: DataEnvelopeSpec()
File: test/scala/akka/cluster/ddata/DataEnvelopeSpec.scala:18
Taint Flags:

```
15  import PruningState._
16
17  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
18  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
19  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
20  val node4 = UniqueAddress(node1.address.copy(port = Some(2554)), 4L)
21  val obsoleteTimeInFuture = System.currentTimeMillis() + 3600 * 1000
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/DataEnvelopeSpec.scala, line 19 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: DataEnvelopeSpec()
File: test/scala/akka/cluster/ddata/DataEnvelopeSpec.scala:19
Taint Flags:

```
16
17  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
18  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
19  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
20  val node4 = UniqueAddress(node1.address.copy(port = Some(2554)), 4L)
21  val obsoleteTimeInFuture = System.currentTimeMillis() + 3600 * 1000
22  val oldObsoleteTime = System.currentTimeMillis() - 3600 * 1000
```

test/scala/akka/cluster/ddata/DataEnvelopeSpec.scala, line 20 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: DataEnvelopeSpec()
File: test/scala/akka/cluster/ddata/DataEnvelopeSpec.scala:20
Taint Flags:

```
17  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
18  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
19  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
20  val node4 = UniqueAddress(node1.address.copy(port = Some(2554)), 4L)
21  val obsoleteTimeInFuture = System.currentTimeMillis() + 3600 * 1000
22  val oldObsoleteTime = System.currentTimeMillis() - 3600 * 1000
23
```

test/scala/akka/cluster/ddata/VersionVectorSpec.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/VersionVectorSpec.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: node1
Enclosing Method: VersionVectorSpec()
File: test/scala/akka/cluster/ddata/VersionVectorSpec.scala:23
Taint Flags:

```
20  with BeforeAndAfterAll {
21
22  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
23  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
24  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
25  val node4 = UniqueAddress(node1.address.copy(port = Some(2554)), 4L)
26
```

test/scala/akka/cluster/ddata/VersionVectorSpec.scala, line 24 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: VersionVectorSpec()
File: test/scala/akka/cluster/ddata/VersionVectorSpec.scala:24
Taint Flags:

```
21
22  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
23  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
24  val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
25  val node4 = UniqueAddress(node1.address.copy(port = Some(2554)), 4L)
26
27  override def afterAll(): Unit = {
```

test/scala/akka/cluster/ddata/VersionVectorSpec.scala, line 25 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: VersionVectorSpec()

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/VersionVectorSpec.scala, line 25 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: test/scala/akka/cluster/ddata/VersionVectorSpec.scala:25
Taint Flags:

```
22 val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
23 val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
24 val node3 = UniqueAddress(node1.address.copy(port = Some(2553)), 3L)
25 val node4 = UniqueAddress(node1.address.copy(port = Some(2554)), 4L)
26
27 override def afterAll(): Unit = {
28 shutdown()
```

test/scala/akka/cluster/ddata/LotsOfDataBot.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: isPassive
Enclosing Method: LotsOfDataBot()
File: test/scala/akka/cluster/ddata/LotsOfDataBot.scala:79
Taint Flags:

```
76 import context.dispatcher
77 val isPassive = context.system.settings.config.getBoolean("passive")
78 var tickTask =
79 if (isPassive)
80 context.system.scheduler.scheduleWithFixedDelay(1.seconds, 1.seconds, self, Tick)
81 else
82 context.system.scheduler.scheduleWithFixedDelay(20.millis, 20.millis, self, Tick)
```

main/scala/akka/cluster/ddata/VersionVector.scala, line 22 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: emptyVersions
Enclosing Method: VersionVector()
File: main/scala/akka/cluster/ddata/VersionVector.scala:22
Taint Flags:

```
19 object VersionVector {
```

Code Correctness: Constructor Invokes Overridable Function Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/VersionVector.scala, line 22 (Code Correctness: Constructor Invokes Overridable Function) Low

```
20
21  private val emptyVersions: TreeMap[UniqueAddress, Long] = TreeMap.empty
22  val empty: VersionVector = ManyVersionVector(emptyVersions)
23
24  def apply(): VersionVector = empty
25
```

main/scala/akka/cluster/ddata/ORMap.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function) Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: $default$4
Enclosing Method: ORMap()
File: main/scala/akka/cluster/ddata/ORMap.scala:16
Taint Flags:

```
13 import akka.util.HashCode
14
15 object ORMap {
16  private val _empty: ORMap[Any, ReplicatedData] = new ORMap(ORSet.empty, Map.empty, VanillaORMapTag)
17  def empty[A, B <: ReplicatedData]: ORMap[A, B] = _empty.asInstanceOf[ORMap[A, B]]
18  def apply(): ORMap[Any, ReplicatedData] = _empty
19
```

test/scala/akka/cluster/ddata/ReplicatorSettingsSpec.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function) Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: config
Enclosing Method: ReplicatorSettingsSpec()
File: test/scala/akka/cluster/ddata/ReplicatorSettingsSpec.scala:23
Taint Flags:

```
20 }
21
22 class ReplicatorSettingsSpec
23  extends AkkaSpec(ReplicatorSettingsSpec.config)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/ReplicatorSettingsSpec.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
24  with AnyWordSpecLike
25  with BeforeAndAfterAll {
26
```

test/scala/akka/cluster/ddata/PNCounterSpec.scala, line 16 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node1
Enclosing Method: PNCounterSpec()
File: test/scala/akka/cluster/ddata/PNCounterSpec.scala:16
Taint Flags:

```
13
14 class PNCounterSpec extends AnyWordSpec with Matchers {
15  val node1 = UniqueAddress(Address("akka", "Sys", "localhost", 2551), 1L)
16  val node2 = UniqueAddress(node1.address.copy(port = Some(2552)), 2L)
17
18  "A PNCounter" must {
19
```

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 155 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: timeout
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:155
Taint Flags:

```
152
153  val data = GSet.empty + "A" + "B"
154  val timeout = 3.seconds.dilated
155  val writeThree = WriteTo(3, timeout)
156  val writeMajority = WriteMajority(timeout)
157  val writeAll = WriteAll(timeout)
158
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function | Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 156 (Code Correctness: Constructor Invokes Overridable Function) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: timeout
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:156
Taint Flags:

```
153  val data = GSet.empty + "A" + "B"
154  val timeout = 3.seconds.dilated
155  val writeThree = WriteTo(3, timeout)
156  val writeMajority = WriteMajority(timeout)
157  val writeAll = WriteAll(timeout)
158
159  val selfUniqueAddress: UniqueAddress = Cluster(system).selfUniqueAddress
```

test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala, line 157 (Code Correctness: Constructor Invokes Overridable Function) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: timeout
Enclosing Method: WriteAggregatorSpec()
File: test/scala/akka/cluster/ddata/WriteAggregatorSpec.scala:157
Taint Flags:

```
154  val timeout = 3.seconds.dilated
155  val writeThree = WriteTo(3, timeout)
156  val writeMajority = WriteMajority(timeout)
157  val writeAll = WriteAll(timeout)
158
159  val selfUniqueAddress: UniqueAddress = Cluster(system).selfUniqueAddress
160
```

main/scala/akka/cluster/ddata/DistributedData.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/DistributedData.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: isTerminated
Enclosing Method: DistributedData()
File: main/scala/akka/cluster/ddata/DistributedData.scala:42
Taint Flags:

```
39 * `ActorRef` of the [[Replicator]] .
40 */
41 val replicator: ActorRef =
42 if (isTerminated) {
43 val log = Logging(system, classOf[DistributedData])
44 if (Cluster(system).isTerminated)
45 log.warning("Replicator points to dead letters, because Cluster is terminated.")
```

main/scala/akka/cluster/ddata/Replicator.scala, line 1445 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: deltaPropagationSelector
Enclosing Method: Replicator()
File: main/scala/akka/cluster/ddata/Replicator.scala:1445
Taint Flags:

```
1442 // Derive the deltaPropagationInterval from the gossipInterval.
1443 // Normally the delta is propagated to all nodes within the gossip tick, so that
1444 // full state gossip is not needed.
1445 val deltaPropagationInterval = (gossipInterval / deltaPropagationSelector.gossipIntervalDivisor).max(200.millis)
1446 Some(
1447 context.system.scheduler
1448 .scheduleWithFixedDelay(deltaPropagationInterval, deltaPropagationInterval, self, DeltaPropagationTick))
```

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: WriteNackManifest

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetSuccessManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: NotFoundManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GCounterKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: PNCounterMapManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: SubscribeManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetFailureManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: PNCounterKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORMultiMapKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 160 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cacheTimeToLive
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:160
Taint Flags:

```
157 private val cacheTimeToLive = system.settings.config
158 .getDuration("akka.cluster.distributed-data.serializer-cache-time-to-live", TimeUnit.MILLISECONDS)
159 .millis
160 private val readCache = new SmallCache[Read, Array[Byte]](4, cacheTimeToLive, m => readToProto(m).toByteArray)
161 private val writeCache = new SmallCache[Write, Array[Byte]](4, cacheTimeToLive, m => writeToProto(m).toByteArray)
162 system.scheduler.scheduleWithFixedDelay(cacheTimeToLive, cacheTimeToLive / 2) { () =>
163 readCache.evict()
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 161 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cacheTimeToLive
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:161
Taint Flags:

```
158 .getDuration("akka.cluster.distributed-data.serializer-cache-time-to-live", TimeUnit.MILLISECONDS)
159 .millis
160 private val readCache = new SmallCache[Read, Array[Byte]](4, cacheTimeToLive, m => readToProto(m).toByteArray)
161 private val writeCache = new SmallCache[Write, Array[Byte]](4, cacheTimeToLive, m => writeToProto(m).toByteArray)
162 system.scheduler.scheduleWithFixedDelay(cacheTimeToLive, cacheTimeToLive / 2) { () =>
163 readCache.evict()
164 writeCache.evict()
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 165 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 165 (Code Correctness: Constructor Invokes Overridable Function)

Low

Sink Details

Sink: FunctionCall: cacheTimeToLive
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:165
Taint Flags:

```
162 system.scheduler.scheduleWithFixedDelay(cacheTimeToLive, cacheTimeToLive / 2) { () =>
163 readCache.evict()
164 writeCache.evict()
165 }(system.dispatchers.internalDispatcher)
166
167 private val writeAckBytes = dm.Empty.getDefaultInstance.toByteArray
168 private val dummyAddress = UniqueAddress(Address("a", "b", "c", 2552), 1L)
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 165 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cacheTimeToLive
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:165
Taint Flags:

```
162 system.scheduler.scheduleWithFixedDelay(cacheTimeToLive, cacheTimeToLive / 2) { () =>
163 readCache.evict()
164 writeCache.evict()
165 }(system.dispatchers.internalDispatcher)
166
167 private val writeAckBytes = dm.Empty.getDefaultInstance.toByteArray
168 private val dummyAddress = UniqueAddress(Address("a", "b", "c", 2552), 1L)
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORMapPutManifest
Enclosing Method: ReplicatedDataSerializer()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORSetManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORMapDeltaGroupManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function)

Low

```
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORSetDeltaGroupManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: DeltaNackManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
190  GetManifest -> getFromBinary,
191  GetSuccessManifest -> getSuccessFromBinary,
192  NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GSetKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORMultiMapManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: LWWRegisterManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: Protocol
Enclosing Method: ReplicatedDataSerializerSpec()
File: test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala:46
Taint Flags:

```
43
44 val Protocol = if (RARP(system).provider.remoteSettings.Artery.Enabled) "akka" else "akka.tcp"
45
46 val address1 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4711), 1L)
47 val address2 = UniqueAddress(Address(Protocol, system.name, "other.host.org", 4711), 2L)
48 val address3 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4712), 3L)
49
```

test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala, line 47 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala, line 47 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: Protocol
Enclosing Method: ReplicatedDataSerializerSpec()
File: test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala:47
Taint Flags:

```
44  val Protocol = if (RARP(system).provider.remoteSettings.Artery.Enabled) "akka" else "akka.tcp"
45
46  val address1 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4711), 1L)
47  val address2 = UniqueAddress(Address(Protocol, system.name, "other.host.org", 4711), 2L)
48  val address3 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4712), 3L)
49
50  val ref1 = system.actorOf(Props.empty, "ref1")
```

test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: Protocol
Enclosing Method: ReplicatedDataSerializerSpec()
File: test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala:48
Taint Flags:

```
45
46  val address1 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4711), 1L)
47  val address2 = UniqueAddress(Address(Protocol, system.name, "other.host.org", 4711), 2L)
48  val address3 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4712), 3L)
49
50  val ref1 = system.actorOf(Props.empty, "ref1")
51  val ref2 = system.actorOf(Props.empty, "ref2")
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ReadManifest
Enclosing Method: ReplicatorMessageSerializer()

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: StatusManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: LWWMapKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORSetKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: WriteAckManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186  val DeltaPropagationManifest = "Q"
187  val DeltaNackManifest = "R"
188
189  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: DeltaPropagationManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

test/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializerSpec.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: Protocol
Enclosing Method: ReplicatorMessageSerializerSpec()
File: test/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializerSpec.scala:51
Taint Flags:

```
48
49 val Protocol = if (RARP(system).provider.remoteSettings.Artery.Enabled) "akka" else "akka.tcp"
50
51 val address1 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4711), 1L)
52 val address2 = UniqueAddress(Address(Protocol, system.name, "other.host.org", 4711), 2L)
53 val address3 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4712), 3L)
54
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

test/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializerSpec.scala, line 52 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: Protocol
Enclosing Method: ReplicatorMessageSerializerSpec()
File: test/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializerSpec.scala:52
Taint Flags:

```
49  val Protocol = if (RARP(system).provider.remoteSettings.Artery.Enabled) "akka" else "akka.tcp"
50
51  val address1 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4711), 1L)
52  val address2 = UniqueAddress(Address(Protocol, system.name, "other.host.org", 4711), 2L)
53  val address3 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4712), 3L)
54
55  val keyA = GSetKey[String]("A")
```

test/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializerSpec.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: Protocol
Enclosing Method: ReplicatorMessageSerializerSpec()
File: test/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializerSpec.scala:53
Taint Flags:

```
50
51  val address1 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4711), 1L)
52  val address2 = UniqueAddress(Address(Protocol, system.name, "other.host.org", 4711), 2L)
53  val address3 = UniqueAddress(Address(Protocol, system.name, "some.host.org", 4712), 3L)
54
55  val keyA = GSetKey[String]("A")
56
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: GossipManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: PNCounterMapKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ReadResultManifest
Enclosing Method: ReplicatorMessageSerializer()

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORSetRemoveManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: DeletedDataManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORMapManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GSetManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function)

Low

```
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: UnsubscribeManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORSetAddManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: DataEnvelopeManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORMapKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: ORMapUpdateManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: WriteManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORSetFullManifest
Enclosing Method: ReplicatedDataSerializer()

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORMapRemoveManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: VersionVectorManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: FlagKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: LWWMapManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function)

Low

```
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ChangedManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ORMapRemoveKeyManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: PNCounterManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296  private val ORMultiMapKeyManifest = "k"
297  private val VersionVectorManifest = "L"
298
299  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300  GSetManifest -> gsetFromBinary,
301  ORSetManifest -> orsetFromBinary,
302  ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function | Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala, line 189 (Code Correctness: Constructor Invokes Overridable Function) | Low

Sink Details

Sink: FunctionCall: DurableDataEnvelopeManifest
Enclosing Method: ReplicatorMessageSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializer.scala:189
Taint Flags:

```
186 val DeltaPropagationManifest = "Q"
187 val DeltaNackManifest = "R"
188
189 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
190 GetManifest -> getFromBinary,
191 GetSuccessManifest -> getSuccessFromBinary,
192 NotFoundManifest -> notFoundFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: FlagManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: LWWRegisterKeyManifest
Enclosing Method: ReplicatedDataSerializer()

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.ddata.protobuf

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala, line 299 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GCounterManifest
Enclosing Method: ReplicatedDataSerializer()
File: main/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializer.scala:299
Taint Flags:

```
296 private val ORMultiMapKeyManifest = "k"
297 private val VersionVectorManifest = "L"
298
299 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
300 GSetManifest -> gsetFromBinary,
301 ORSetManifest -> orsetFromBinary,
302 ORSetAddManifest -> orsetAddFromBinary,
```

Code Correctness: Erroneous String Compare (4 issues)

Abstract

Strings should be compared with the `equals()` method, not `==` or `!=`.

Explanation

This program uses `==` or `!=` to compare two strings for equality, which compares two objects for equality, not their values. Chances are good that the two references will never be equal. **Example 1:** The following branch will never be taken.

```
  if (args[0] == STRING_CONSTANT) {
      logger.info("miracle");
  }
```

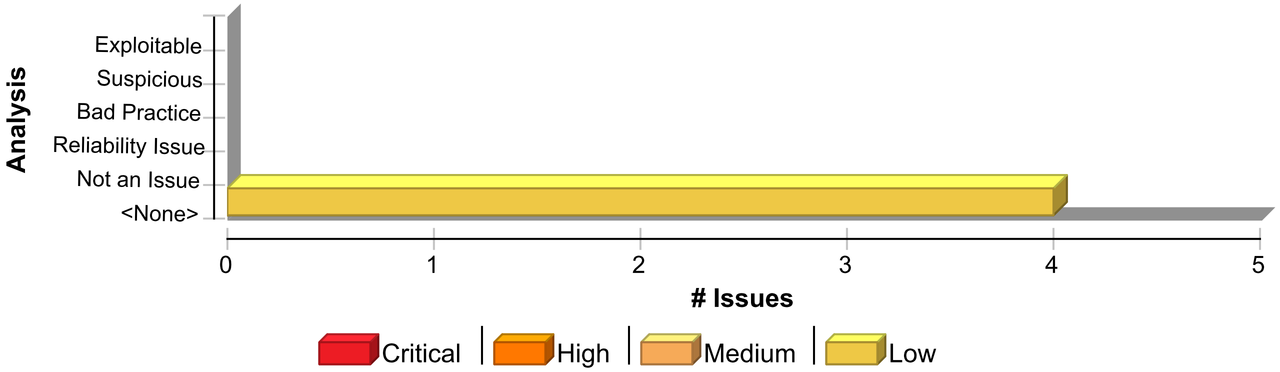
The `==` and `!=` operators will only behave as expected when they are used to compare strings contained in objects that are equal. The most common way for this to occur is for the strings to be interned, whereby the strings are added to a pool of objects maintained by the `String` class. Once a string is interned, all uses of that string will use the same object and equality operators will behave as expected. All string literals and string-valued constants are interned automatically. Other strings can be interned manually be calling `String.intern()`, which will return a canonical instance of the current string, creating one if necessary.

Recommendation

Use `equals()` to compare strings. **Example 2:** The code in `Example 1` could be rewritten in the following way:

```
  if (STRING_CONSTANT.equals(args[0])) {
      logger.info("could happen");
  }
```

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Erroneous String Compare	4	0	0	4
Total	**4**	**0**	**0**	**4**

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Erroneous String Compare

Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 375 (Code Correctness: Erroneous String Compare)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: withDispatcher()
File: main/scala/akka/cluster/ddata/Replicator.scala:375
Taint Flags:

```
372 copy(maxDeltaElements = maxDeltaElements)
373
374 def withDispatcher(dispatcher: String): ReplicatorSettings = {
375 val d = dispatcher match {
376 case "" => Dispatchers.InternalDispatcherId
377 case id => id
378 }
```

main/scala/akka/cluster/ddata/Replicator.scala, line 81 (Code Correctness: Erroneous String Compare)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: main/scala/akka/cluster/ddata/Replicator.scala:81
Taint Flags:

```
78 def apply(config: Config): ReplicatorSettings = {
79 val dispatcher = config.getString("use-dispatcher")
80
81 val pruningInterval = toRootLowerCase(config.getString("pruning-interval")) match {
82 case "off" | "false" => Duration.Zero
83 case _ => config.getDuration("pruning-interval", MILLISECONDS).millis
84 }
```

main/scala/akka/cluster/ddata/DurableStore.scala, line 123 (Code Correctness: Erroneous String Compare)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/DurableStore.scala, line 123 (Code Correctness: Erroneous String Compare) — Low

Sink Details

Sink: Operation
Enclosing Method: LmdbDurableStore()
File: main/scala/akka/cluster/ddata/DurableStore.scala:123
Taint Flags:

```
120 val serializer = serialization.serializerFor(classOf[DurableDataEnvelope]).asInstanceOf[SerializerWithStringManifest]
121 val manifest = serializer.manifest(new DurableDataEnvelope(Replicator.Internal.DeletedData))
122
123 val writeBehindInterval = config.getString("lmdb.write-behind-interval").toLowerCase match {
124 case "off" => Duration.Zero
125 case _ => config.getDuration("lmdb.write-behind-interval", MILLISECONDS).millis
126 }
```

main/scala/akka/cluster/ddata/Replicator.scala, line 81 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: main/scala/akka/cluster/ddata/Replicator.scala:81
Taint Flags:

```
78 def apply(config: Config): ReplicatorSettings = {
79 val dispatcher = config.getString("use-dispatcher")
80
81 val pruningInterval = toRootLowerCase(config.getString("pruning-interval")) match {
82 case "off" | "false" => Duration.Zero
83 case _ => config.getDuration("pruning-interval", MILLISECONDS).millis
84 }
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable (45 issues)

Abstract

Inner classes implementing `java.io.Serializable` may cause problems and leak information from the outer class.

Explanation

Serialization of inner classes lead to serialization of the outer class, therefore possibly leaking information or leading to a runtime error if the outer class is not serializable. As well as this, serializing inner classes may cause platform dependencies since the Java compiler creates synthetic fields in order to implement inner classes, but these are implementation dependent, and may vary from compiler to compiler. **Example 1:** The following code allows serialization of an inner class.

```
...
class User implements Serializable {
  private int accessLevel;
  class Registrator implements Serializable {
    ...
  }
}
```

In `Example 1`, when the inner class `Registrator` is serialized, it will also serialize the field `accessLevel` from the outer class `User`.

Recommendation

When using inner classes, they should not be serialized, or they should be changed to static-nested classes, since these do not have the drawbacks that non-static inner classes have when serialized. When a nested class is static it inherently has no association with instance variables (including those of the outer class), and would not cause serialization of the outer class. **Example 2:** The following code changes the example in `Example 1`, by stopping the inner class from implementing `java.io.Serializable`.

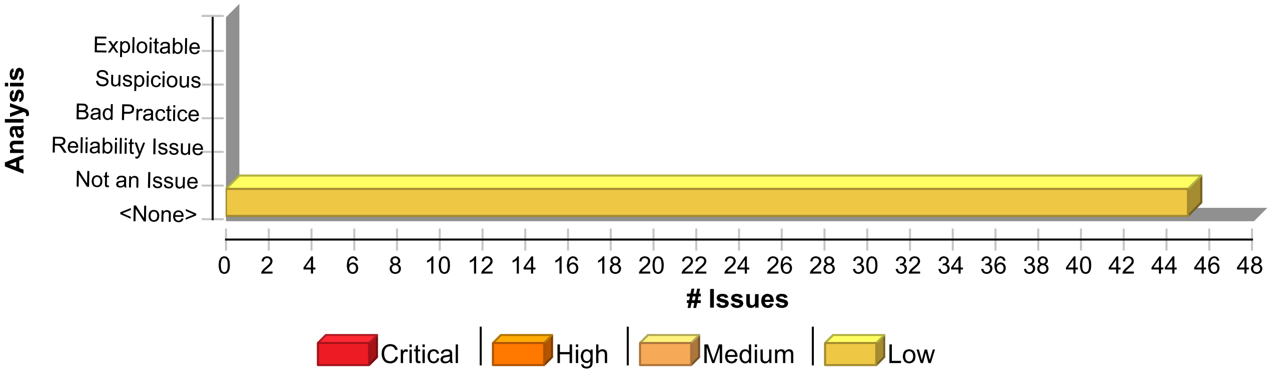
```
...
class User implements Serializable {
  private int accessLevel;
  class Registrator {
    ...
  }
}
```

Example 2: The following code changes the example in `Example 1`, by making the inner class into a static-nested class.

```
...
class User implements Serializable {
  private int accessLevel;
  static class Registrator implements Serializable {
    ...
  }
}
```

Issue Summary

© Copyright [2008-2022] Micro Focus or one of its affiliates.



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Non-Static Inner Class Implements Serializable	45	0	0	45
Total	**45**	**0**	**0**	**45**

Code Correctness: Non-Static Inner Class Implements Serializable

Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 850 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$StoreFailure
File: main/scala/akka/cluster/ddata/Replicator.scala:850
Taint Flags:

```
847 * It will eventually be disseminated to other replicas, unless the local replica
848 * crashes before it has been able to communicate with other replicas.
849 */
850 final case class StoreFailure[A <: ReplicatedData](key: Key[A], request: Option[Any])
851 extends UpdateFailure[A]
852 with DeleteResponse[A] {
853
```

main/scala/akka/cluster/ddata/Replicator.scala, line 626 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$GetSuccess

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 626 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

File: main/scala/akka/cluster/ddata/Replicator.scala:626
Taint Flags:

```
623 /**
624 * Reply from `Get`. The data value is retrieved with [[#get]] using the typed key.
625 */
626 final case class GetSuccess[A <: ReplicatedData](key: Key[A], request: Option[Any])(data: A)
627 extends GetResponse[A]
628 with ReplicatorMessage {
629
```

main/scala/akka/cluster/ddata/Replicator.scala, line 718 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$Deleted
File: main/scala/akka/cluster/ddata/Replicator.scala:718
Taint Flags:

```
715 /**
716 * @see [[Replicator.Subscribe]]
717 */
718 final case class Deleted[A <: ReplicatedData](key: Key[A]) extends SubscribeResponse[A]
719
720 object Update {
721
```

main/scala/akka/cluster/ddata/Replicator.scala, line 644 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$NotFound
File: main/scala/akka/cluster/ddata/Replicator.scala:644
Taint Flags:

```
641 */
642 def dataValue: A = data
643 }
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 644 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
644 final case class NotFound[A <: ReplicatedData](key: Key[A], request: Option[Any])
645 extends GetResponse[A]
646 with ReplicatorMessage
647
```

main/scala/akka/cluster/ddata/Replicator.scala, line 675 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$Subscribe
File: main/scala/akka/cluster/ddata/Replicator.scala:675
Taint Flags:

```
672 * If the key is deleted the subscriber is notified with a [[Deleted]]
673 * message.
674 */
675 final case class Subscribe[A <: ReplicatedData](key: Key[A], subscriber: ActorRef) extends ReplicatorMessage
676
677 /**
678 * Unregister a subscriber.
```

main/scala/akka/cluster/ddata/Replicator.scala, line 483 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$ReadMajority
File: main/scala/akka/cluster/ddata/Replicator.scala:483
Taint Flags:

```
480 */
481 def this(n: Int, timeout: java.time.Duration) = this(n, timeout.asScala)
482 }
483 final case class ReadMajority(timeout: FiniteDuration, minCap: Int = DefaultMajorityMinCap) extends ReadConsistency {
484 def this(timeout: FiniteDuration) = this(timeout, DefaultMajorityMinCap)
485
486 /**
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable | Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 519 (Code Correctness: Non-Static Inner Class Implements Serializable) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$WriteTo
File: main/scala/akka/cluster/ddata/Replicator.scala:519
Taint Flags:

```
516  case object WriteLocal extends WriteConsistency {
517  override def timeout: FiniteDuration = Duration.Zero
518  }
519  final case class WriteTo(n: Int, timeout: FiniteDuration) extends WriteConsistency {
520  require(n >= 2, "WriteTo n must be >= 2, use WriteLocal for n=1")
521
522  /**
```

main/scala/akka/cluster/ddata/Replicator.scala, line 542 (Code Correctness: Non-Static Inner Class Implements Serializable) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$WriteMajorityPlus
File: main/scala/akka/cluster/ddata/Replicator.scala:542
Taint Flags:

```
539  * all nodes. Exiting nodes are excluded using `WriteMajorityPlus` because those are typically
540  * about to be removed and will not be able to respond.
541  */
542  final case class WriteMajorityPlus(timeout: FiniteDuration, additional: Int, minCap: Int = DefaultMajorityMinCap)
543  extends WriteConsistency {
544
545  /**
```

main/scala/akka/cluster/ddata/DurableStore.scala, line 107 (Code Correctness: Non-Static Inner Class Implements Serializable) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/DurableStore.scala, line 107 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: LmdbDurableStore$Lmdb
File: main/scala/akka/cluster/ddata/DurableStore.scala:107
Taint Flags:

```
104
105  private case object WriteBehind extends DeadLetterSuppression
106
107  private final case class Lmdb(
108  env: Env[ByteBuffer],
109  db: Dbi[ByteBuffer],
110  keyBuffer: ByteBuffer,
```

main/scala/akka/cluster/ddata/Replicator.scala, line 527 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$WriteMajority
File: main/scala/akka/cluster/ddata/Replicator.scala:527
Taint Flags:

```
524  */
525  def this(n: Int, timeout: java.time.Duration) = this(n, timeout.asScala)
526  }
527  final case class WriteMajority(timeout: FiniteDuration, minCap: Int = DefaultMajorityMinCap)
528  extends WriteConsistency {
529  def this(timeout: FiniteDuration) = this(timeout, DefaultMajorityMinCap)
530
```

main/scala/akka/cluster/ddata/Replicator.scala, line 505 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$ReadAll
File: main/scala/akka/cluster/ddata/Replicator.scala:505
Taint Flags:

```
502  */
503  def this(timeout: java.time.Duration, additional: Int) = this(timeout.asScala, additional, DefaultMajorityMinCap)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 505 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
504  }
505  final case class ReadAll(timeout: FiniteDuration) extends ReadConsistency {
506
507  /**
508  * Java API
```

main/scala/akka/cluster/ddata/Replicator.scala, line 576 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$GetKeyIdsResult
File: main/scala/akka/cluster/ddata/Replicator.scala:576
Taint Flags:

```
573  /**
574  * INTERNAL API
575  */
576  @InternalApi private[akka] final case class GetKeyIdsResult(keyIds: Set[KeyId]) {
577
578  /**
579  * Java API
```

test/scala/akka/cluster/ddata/LocalConcurrencySpec.scala, line 21 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: LocalConcurrencySpec$Add
File: test/scala/akka/cluster/ddata/LocalConcurrencySpec.scala:21
Taint Flags:

```
18
19 object LocalConcurrencySpec {
20
21  final case class Add(s: String)
22
23  object Updater {
24  val key = ORSetKey[String]("key")
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/LocalConcurrencySpec.scala, line 21 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

main/scala/akka/cluster/ddata/Replicator.scala, line 550 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$WriteAll
File: main/scala/akka/cluster/ddata/Replicator.scala:550
Taint Flags:

```
547 */
548 def this(timeout: java.time.Duration, additional: Int) = this(timeout.asScala, additional, DefaultMajorityMinCap)
549 }
550 final case class WriteAll(timeout: FiniteDuration) extends WriteConsistency {
551
552 /**
553 * Java API
```

main/scala/akka/cluster/ddata/Replicator.scala, line 884 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$ReplicationDeleteFailure
File: main/scala/akka/cluster/ddata/Replicator.scala:884
Taint Flags:

```
881 def getRequest: Optional[Any] = Optional.ofNullable(request.orNull)
882 }
883 final case class DeleteSuccess[A <: ReplicatedData](key: Key[A], request: Option[Any]) extends DeleteResponse[A]
884 final case class ReplicationDeleteFailure[A <: ReplicatedData](key: Key[A], request: Option[Any])
885 extends DeleteResponse[A]
886 final case class DataDeleted[A <: ReplicatedData](key: Key[A], request: Option[Any])
887 extends RuntimeException
```

main/scala/akka/cluster/ddata/ORSet.scala, line 99 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable

Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/ORSet.scala, line 99 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORSet$DeltaGroup
File: main/scala/akka/cluster/ddata/ORSet.scala:99
Taint Flags:

```
96  /**
97  * INTERNAL API
98  */
99  @InternalApi private[akka] final case class DeltaGroup[A](ops: immutable.IndexedSeq[DeltaOp])
100  extends DeltaOp
101  with ReplicatedDeltaSize {
102  override def merge(that: DeltaOp): DeltaOp = that match {
```

main/scala/akka/cluster/ddata/ORMap.scala, line 56 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORMap$AtomicDeltaOp
File: main/scala/akka/cluster/ddata/ORMap.scala:56
Taint Flags:

```
53  /**
54  * INTERNAL API
55  */
56  @InternalApi private[akka] sealed abstract class AtomicDeltaOp[A, B <: ReplicatedData]
57  extends DeltaOp
58  with ReplicatedDeltaSize {
59  def underlying: ORSet.DeltaOp
```

main/scala/akka/cluster/ddata/Replicator.scala, line 831 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable | Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 831 (Code Correctness: Non-Static Inner Class Implements Serializable) | Low

Sink: Class: Replicator$ModifyFailure
File: main/scala/akka/cluster/ddata/Replicator.scala:831
Taint Flags:

```
828  * If the `modify` function of the [[Update]] throws an exception the reply message
829  * will be this `ModifyFailure` message. The original exception is included as `cause`.
830  */
831  final case class ModifyFailure[A <: ReplicatedData](
832  key: Key[A],
833  errorMessage: String,
834  cause: Throwable,
```

main/scala/akka/cluster/ddata/ORMap.scala, line 126 (Code Correctness: Non-Static Inner Class Implements Serializable) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORMap$RemoveDeltaOp
File: main/scala/akka/cluster/ddata/ORMap.scala:126
Taint Flags:

```
123
124  // RemoveDeltaOp does not contain any value at all - the propagated 'value' map would be empty
125  /** INTERNAL API */
126  @InternalApi private[akka] final case class RemoveDeltaOp[A, B <: ReplicatedData](
127  underlying: ORSet.DeltaOp,
128  zeroTag: ZeroTag)
129  extends AtomicDeltaOp[A, B]
```

main/scala/akka/cluster/ddata/Replicator.scala, line 820 (Code Correctness: Non-Static Inner Class Implements Serializable) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$UpdateTimeout
File: main/scala/akka/cluster/ddata/Replicator.scala:820
Taint Flags:

```
817  * It will eventually be disseminated to other replicas, unless the local replica
818  * crashes before it has been able to communicate with other replicas.
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 820 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
819 */
820 final case class UpdateTimeout[A <: ReplicatedData](key: Key[A], request: Option[Any]) extends UpdateFailure[A]
821
822 /**
823 * The [[Update]] couldn't be performed because the entry has been deleted.
```

main/scala/akka/cluster/ddata/Replicator.scala, line 886 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$DataDeleted
File: main/scala/akka/cluster/ddata/Replicator.scala:886
Taint Flags:

```
883 final case class DeleteSuccess[A <: ReplicatedData](key: Key[A], request: Option[Any]) extends DeleteResponse[A]
884 final case class ReplicationDeleteFailure[A <: ReplicatedData](key: Key[A], request: Option[Any])
885 extends DeleteResponse[A]
886 final case class DataDeleted[A <: ReplicatedData](key: Key[A], request: Option[Any])
887 extends RuntimeException
888 with NoStackTrace
889 with DeleteResponse[A] {
```

main/scala/akka/cluster/ddata/Replicator.scala, line 652 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$GetFailure
File: main/scala/akka/cluster/ddata/Replicator.scala:652
Taint Flags:

```
649 * The [[Get]] request could not be fulfill according to the given
650 * [[ReadConsistency consistency level]] and [[ReadConsistency#timeout timeout]].
651 */
652 final case class GetFailure[A <: ReplicatedData](key: Key[A], request: Option[Any])
653 extends GetResponse[A]
654 with ReplicatorMessage
655
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 652 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

main/scala/akka/cluster/ddata/Replicator.scala, line 883 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$DeleteSuccess
File: main/scala/akka/cluster/ddata/Replicator.scala:883
Taint Flags:

```
880 /** Java API*/
881 def getRequest: Optional[Any] = Optional.ofNullable(request.orNull)
882 }
883 final case class DeleteSuccess[A <: ReplicatedData](key: Key[A], request: Option[Any]) extends DeleteResponse[A]
884 final case class ReplicationDeleteFailure[A <: ReplicatedData](key: Key[A], request: Option[Any])
885 extends DeleteResponse[A]
886 final case class DataDeleted[A <: ReplicatedData](key: Key[A], request: Option[Any])
```

main/scala/akka/cluster/ddata/ORMap.scala, line 141 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORMap$DeltaGroup
File: main/scala/akka/cluster/ddata/ORMap.scala:141
Taint Flags:

```
138
139 // DeltaGroup is effectively a causally ordered list of individual deltas
140 /** INTERNAL API */
141 @InternalApi private[akka] final case class DeltaGroup[A, B <: ReplicatedData](ops: immutable.IndexedSeq[DeltaOp])
142 extends DeltaOp
143 with ReplicatedDeltaSize {
144 override def merge(that: DeltaOp): DeltaOp = that match {
```

main/scala/akka/cluster/ddata/Replicator.scala, line 497 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 497 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$ReadMajorityPlus
File: main/scala/akka/cluster/ddata/Replicator.scala:497
Taint Flags:

```
494 * all nodes. Exiting nodes are excluded using `ReadMajorityPlus` because those are typically
495 * about to be removed and will not be able to respond.
496 */
497 final case class ReadMajorityPlus(timeout: FiniteDuration, additional: Int, minCap: Int = DefaultMajorityMinCap)
498 extends ReadConsistency {
499
500 /**
```

main/scala/akka/cluster/ddata/Replicator.scala, line 696 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$Changed
File: main/scala/akka/cluster/ddata/Replicator.scala:696
Taint Flags:

```
693 *
694 * @see [[Replicator.Subscribe]]
695 */
696 final case class Changed[A <: ReplicatedData](key: Key[A])(data: A)
697 extends SubscribeResponse[A]
698 with ReplicatorMessage {
699
```

main/scala/akka/cluster/ddata/Replicator.scala, line 475 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 475 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: Replicator$ReadFrom
File: main/scala/akka/cluster/ddata/Replicator.scala:475
Taint Flags:

```
472 case object ReadLocal extends ReadConsistency {
473 override def timeout: FiniteDuration = Duration.Zero
474 }
475 final case class ReadFrom(n: Int, timeout: FiniteDuration) extends ReadConsistency {
476 require(n >= 2, "ReadFrom n must be >= 2, use ReadLocal for n=1")
477
478 /**
```

main/scala/akka/cluster/ddata/PruningState.scala, line 24 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: PruningState$PruningPerformed
File: main/scala/akka/cluster/ddata/PruningState.scala:24
Taint Flags:

```
21 }
22 def estimatedSize: Int = EstimatedSize.UniqueAddress + EstimatedSize.Address * seen.size
23 }
24 final case class PruningPerformed(obsoleteTime: Long) extends PruningState {
25 def isObsolete(currentTime: Long): Boolean = obsoleteTime <= currentTime
26 def addSeen(@unused node: Address): PruningState = this
27 def estimatedSize: Int = EstimatedSize.LongValue
```

main/scala/akka/cluster/ddata/PruningState.scala, line 17 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: PruningState$PruningInitialized
File: main/scala/akka/cluster/ddata/PruningState.scala:17
Taint Flags:

```
14 * INTERNAL API
15 */
```

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/PruningState.scala, line 17 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
16 @InternalApi private[akka] object PruningState {
17 final case class PruningInitialized(owner: UniqueAddress, seen: Set[Address]) extends PruningState {
18 override def addSeen(node: Address): PruningState = {
19 if (seen(node) || owner.address == node) this
20 else copy(seen = seen + node)
```

main/scala/akka/cluster/ddata/Replicator.scala, line 866 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$Delete
File: main/scala/akka/cluster/ddata/Replicator.scala:866
Taint Flags:

```
863 * way to pass contextual information (e.g. original sender) without having to use `ask`
864 * or maintain local correlation data structures.
865 */
866 final case class Delete[A <: ReplicatedData](key: Key[A], consistency: WriteConsistency, request: Option[Any] = None)
867 extends Command[A]
868 with NoSerializationVerificationNeeded {
869
```

main/scala/akka/cluster/ddata/Replicator.scala, line 682 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$Unsubscribe
File: main/scala/akka/cluster/ddata/Replicator.scala:682
Taint Flags:

```
679 *
680 * @see [[Replicator.Subscribe]]
681 */
682 final case class Unsubscribe[A <: ReplicatedData](key: Key[A], subscriber: ActorRef) extends ReplicatorMessage
683
684 /**
685 * @see [[Replicator.Subscribe]]
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 682 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

main/scala/akka/cluster/ddata/Replicator.scala, line 825 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$UpdateDataDeleted
File: main/scala/akka/cluster/ddata/Replicator.scala:825
Taint Flags:

```
822 /**
823 * The [[Update]] couldn't be performed because the entry has been deleted.
824 */
825 final case class UpdateDataDeleted[A <: ReplicatedData](key: Key[A], request: Option[Any]) extends UpdateResponse[A]
826
827 /**
828 * If the `modify` function of the [[Update]] throws an exception the reply message
```

main/scala/akka/cluster/ddata/ORSet.scala, line 89 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORSet$FullStateDeltaOp
File: main/scala/akka/cluster/ddata/ORSet.scala:89
Taint Flags:

```
86 }
87
88 /** INTERNAL API: Used for `clear` but could be used for other cases also */
89 @InternalApi private[akka] final case class FullStateDeltaOp[A](underlying: ORSet[A]) extends AtomicDeltaOp[A] {
90 override def merge(that: DeltaOp): DeltaOp = that match {
91 case _: AtomicDeltaOp[_] => DeltaGroup(Vector(this, that))
92 case DeltaGroup(ops) => DeltaGroup(this +: ops)
```

main/scala/akka/cluster/ddata/ORSet.scala, line 50 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/ORSet.scala, line 50 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORSet$AtomicDeltaOp
File: main/scala/akka/cluster/ddata/ORSet.scala:50
Taint Flags:

```
47 /**
48 * INTERNAL API
49 */
50 @InternalApi private[akka] sealed abstract class AtomicDeltaOp[A] extends DeltaOp with ReplicatedDeltaSize {
51 def underlying: ORSet[A]
52 override def zero: ORSet[A] = ORSet.empty
53 override def deltaSize: Int = 1
```

main/scala/akka/cluster/ddata/Replicator.scala, line 806 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$UpdateSuccess
File: main/scala/akka/cluster/ddata/Replicator.scala:806
Taint Flags:

```
803 /** Java API */
804 def getRequest: Optional[Any] = Optional.ofNullable(request.orNull)
805 }
806 final case class UpdateSuccess[A <: ReplicatedData](key: Key[A], request: Option[Any])
807 extends UpdateResponse[A]
808 with DeadLetterSuppression
809 sealed abstract class UpdateFailure[A <: ReplicatedData] extends UpdateResponse[A]
```

main/scala/akka/cluster/ddata/ORSet.scala, line 57 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/ORSet.scala, line 57 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: ORSet$AddDeltaOp
File: main/scala/akka/cluster/ddata/ORSet.scala:57
Taint Flags:

```
54 }
55
56 /** INTERNAL API */
57 @InternalApi private[akka] final case class AddDeltaOp[A](underlying: ORSet[A]) extends AtomicDeltaOp[A] {
58
59 override def merge(that: DeltaOp): DeltaOp = that match {
60 case AddDeltaOp(u) =>
```

main/scala/akka/cluster/ddata/Replicator.scala, line 599 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$Get
File: main/scala/akka/cluster/ddata/Replicator.scala:599
Taint Flags:

```
596 * way to pass contextual information (e.g. original sender) without having to use `ask`
597 * or maintain local correlation data structures.
598 */
599 final case class Get[A <: ReplicatedData](key: Key[A], consistency: ReadConsistency, request: Option[Any] = None)
600 extends Command[A]
601 with ReplicatorMessage {
602
```

main/scala/akka/cluster/ddata/ORMap.scala, line 133 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORMap$RemoveKeyDeltaOp
File: main/scala/akka/cluster/ddata/ORMap.scala:133
Taint Flags:

```
130
131 // RemoveKeyDeltaOp contains a single value - to provide the recipient with the removed key for value map
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/ORMap.scala, line 133 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
132 /** INTERNAL API */
133 @InternalApi private[akka] final case class RemoveKeyDeltaOp[A, B <: ReplicatedData](
134 underlying: ORSet.DeltaOp,
135 removedKey: A,
136 zeroTag: ZeroTag)
```

main/scala/akka/cluster/ddata/Replicator.scala, line 762 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$Update
File: main/scala/akka/cluster/ddata/Replicator.scala:762
Taint Flags:

```
759 * function that only uses the data parameter and stable fields from enclosing scope. It must
760 * for example not access `sender()` reference of an enclosing actor.
761 */
762 final case class Update[A <: ReplicatedData](key: Key[A], writeConsistency: WriteConsistency, request: Option[Any])(
763 val modify: Option[A] => A)
764 extends Command[A]
765 with NoSerializationVerificationNeeded {
```

main/scala/akka/cluster/ddata/Replicator.scala, line 659 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$GetDataDeleted
File: main/scala/akka/cluster/ddata/Replicator.scala:659
Taint Flags:

```
656 /**
657 * The [[Get]] request couldn't be performed because the entry has been deleted.
658 */
659 final case class GetDataDeleted[A <: ReplicatedData](key: Key[A], request: Option[Any]) extends GetResponse[A]
660
661 /**
662 * Register a subscriber that will be notified with a [[Changed]] message
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 659 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

main/scala/akka/cluster/ddata/ORMap.scala, line 71 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORMap$PutDeltaOp
File: main/scala/akka/cluster/ddata/ORMap.scala:71
Taint Flags:

```
68
69 // PutDeltaOp contains ORSet delta and full value
70 /** INTERNAL API */
71 @InternalApi private[akka] final case class PutDeltaOp[A, B <: ReplicatedData](
72 underlying: ORSet.DeltaOp,
73 value: (A, B),
74 zeroTag: ZeroTag)
```

main/scala/akka/cluster/ddata/ORMap.scala, line 30 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORMap$DeltaOp
File: main/scala/akka/cluster/ddata/ORMap.scala:30
Taint Flags:

```
27 */
28 def unapply[A, B <: ReplicatedData](m: ORMap[A, B]): Option[Map[A, B]] = Some(m.entries)
29
30 sealed trait DeltaOp extends ReplicatedDelta with RequiresCausalDeliveryOfDeltas with ReplicatedDataSerialization {
31 type T = DeltaOp
32 override def zero: DeltaReplicatedData
33 }
```

main/scala/akka/cluster/ddata/ORSet.scala, line 78 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Code Correctness: Non-Static Inner Class Implements Serializable

Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/ORSet.scala, line 78 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ORSet$RemoveDeltaOp
File: main/scala/akka/cluster/ddata/ORSet.scala:78
Taint Flags:

```
75  }
76
77  /** INTERNAL API */
78  @InternalApi private[akka] final case class RemoveDeltaOp[A](underlying: ORSet[A]) extends AtomicDeltaOp[A] {
79  if (underlying.size != 1)
80  throw new IllegalArgumentException(s"RemoveDeltaOp should contain one removed element, but was $underlying")
81
```

main/scala/akka/cluster/ddata/Replicator.scala, line 907 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Replicator$ReplicaCount
File: main/scala/akka/cluster/ddata/Replicator.scala:907
Taint Flags:

```
904  /**
905  * Current number of replicas. Reply to `GetReplicaCount`.
906  */
907  final case class ReplicaCount(n: Int)
908
909  /**
910  * Notify subscribers of changes now, otherwise they will be notified periodically
```

main/scala/akka/cluster/ddata/ORMap.scala, line 98 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Code Correctness: Non-Static Inner Class Implements Serializable — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/ORMap.scala, line 98 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: ORMap$UpdateDeltaOp
File: main/scala/akka/cluster/ddata/ORMap.scala:98
Taint Flags:

```
95
96  // UpdateDeltaOp contains ORSet delta and either delta of value (in case where underlying type supports deltas) or full value
97  /** INTERNAL API */
98  @InternalApi private[akka] final case class UpdateDeltaOp[A, B <: ReplicatedData](
99  underlying: ORSet.DeltaOp,
100  values: Map[A, B],
101  zeroTag: ZeroTag)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Dead Code: Expression is Always false (11 issues)

Abstract

This expression will always evaluate to `false`.

Explanation

This expression will always evaluate to `false`; the program could be rewritten in a simpler form. The nearby code may be present for debugging purposes, or it may not have been maintained along with the rest of the program. The expression may also be indicative of a bug earlier in the method. **Example 1:** The following method never sets the variable `secondCall` after initializing it to `false`. (The variable `firstCall` is mistakenly used twice.) The result is that the expression `firstCall && secondCall` will always evaluate to `false`, so `setUpDualCall()` will never be invoked.

```
public void setUpCalls() {
  boolean firstCall = false;
  boolean secondCall = false;

  if (fCall > 0) {
    setUpFCall();
    firstCall = true;
  }
  if (sCall > 0) {
    setUpSCall();
    firstCall = true;
  }

  if (firstCall && secondCall) {
    setUpDualCall();
  }
}
```

Example 2: The following method never sets the variable `firstCall` to `true`. (The variable `firstCall` is mistakenly set to `false` after the first conditional statement.) The result is that the first part of the expression `firstCall && secondCall` will always evaluate to `false`.

```
public void setUpCalls() {
  boolean firstCall = false;
  boolean secondCall = false;

  if (fCall > 0) {
    setUpFCall();
    firstCall = false;
  }
  if (sCall > 0) {
    setUpSCall();
    secondCall = true;
  }

  if (firstCall && secondCall) {
    setUpForCall();
  }
}
```

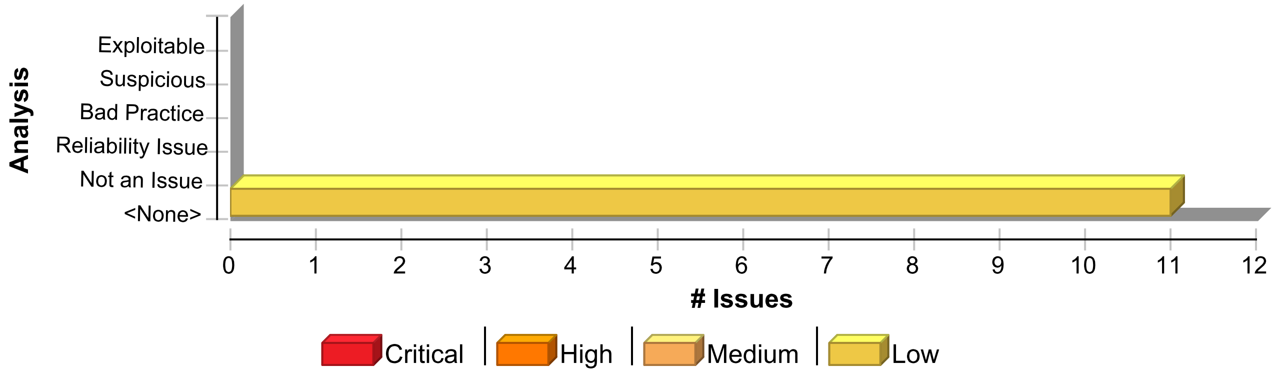
Recommendation

In general, you should repair or remove unused code. It causes additional complexity and maintenance burden without

© Copyright [2008-2022] Micro Focus or one of its affiliates.

contributing to the functionality of the program.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Dead Code: Expression is Always false	11	0	0	11
Total	**11**	**0**	**0**	**11**

Dead Code: Expression is Always false — Low

Package: test.scala.akka.cluster.ddata

test/scala/akka/cluster/ddata/FlagSpec.scala, line 41 (Dead Code: Expression is Always false) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/FlagSpec.scala:41
Taint Flags:

```
38 value2 should be(true)
39
40 Changed(FlagKey("key"))(f1) match {
41 case c @ Changed(FlagKey("key")) =>
42 val Flag(value3) = c.dataValue
43 val value4: Boolean = value3
44 value4 should be(true)
```

test/scala/akka/cluster/ddata/GCounterSpec.scala, line 201 (Dead Code: Expression is Always false) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Dead Code: Expression is Always false — Low

Package: test.scala.akka.cluster.ddata

test/scala/akka/cluster/ddata/GCounterSpec.scala, line 201 (Dead Code: Expression is Always false) — Low

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/GCounterSpec.scala:201
Taint Flags:

```
198 value2 should be(2L)
199
200 Changed(GCounterKey("key"))(c1) match {
201 case c @ Changed(GCounterKey("key")) =>
202 val value3 = c.dataValue match {
203 case GCounter(value3) => value3
204 case _ => fail()
```

test/scala/akka/cluster/ddata/GSetSpec.scala, line 151 (Dead Code: Expression is Always false) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/GSetSpec.scala:151
Taint Flags:

```
148 elements2 should be(Set("a", "b"))
149
150 Changed(GSetKey[String]("key"))(s1) match {
151 case c @ Changed(GSetKey("key")) =>
152 val GSet(elements3) = c.dataValue
153 val elements4: Set[String] = elements3
154 elements4 should be(Set("a", "b"))
```

test/scala/akka/cluster/ddata/LWWMapSpec.scala, line 82 (Dead Code: Expression is Always false) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()

Dead Code: Expression is Always false — Low

Package: test.scala.akka.cluster.ddata

test/scala/akka/cluster/ddata/LWWMapSpec.scala, line 82 (Dead Code: Expression is Always false) — Low

File: test/scala/akka/cluster/ddata/LWWMapSpec.scala:82
Taint Flags:

```
79 entries2 should be(Map("a" -> 1L))
80
81 Changed(LWWMapKey[String, Long]("key"))(m1) match {
82 case c @ Changed(LWWMapKey("key")) =>
83 val entries3 = c.dataValue match {
84 case LWWMap(entries3) => entries3
85 case _ => fail()
```

test/scala/akka/cluster/ddata/LWWRegisterSpec.scala, line 81 (Dead Code: Expression is Always false) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/LWWRegisterSpec.scala:81
Taint Flags:

```
78 value2 should be("a")
79
80 Changed(LWWRegisterKey[String]("key"))(r1) match {
81 case c @ Changed(LWWRegisterKey("key")) =>
82 val value3 = c.dataValue match {
83 case LWWRegister(value3) => value3
84 case _ => fail()
```

test/scala/akka/cluster/ddata/ORMapSpec.scala, line 754 (Dead Code: Expression is Always false) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/ORMapSpec.scala:754
Taint Flags:

```
751 entries2 should be(Map("a" -> Flag(true), "b" -> Flag(false)))
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Dead Code: Expression is Always false

Low

Package: test.scala.akka.cluster.ddata

test/scala/akka/cluster/ddata/ORMapSpec.scala, line 754 (Dead Code: Expression is Always false)

Low

```
752
753  Changed(ORMapKey[String, Flag]("key"))(m1) match {
754  case c @ Changed(ORMapKey("key")) =>
755  val entries3 = c.dataValue match {
756  case ORMap(entries3) => entries3
757  case _ => fail()
```

test/scala/akka/cluster/ddata/ORMultiMapSpec.scala, line 566 (Dead Code: Expression is Always false)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/ORMultiMapSpec.scala:566
Taint Flags:

```
563  entries2 should be(Map("a" -> Set(1L, 2L), "b" -> Set(3L)))
564
565  Changed(ORMultiMapKey[String, Long]("key"))(m1) match {
566  case c @ Changed(ORMultiMapKey("key")) =>
567  val entries3 = c.dataValue match {
568  case ORMultiMap(entries3: Map[String, Set[Long]]) => entries3
569  case _ => fail()
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 616 (Dead Code: Expression is Always false)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:616
Taint Flags:

```
613  elements2 should be(Set("a", "b"))
614
615  Changed(ORSetKey[String]("key"))(s1) match {
616  case c @ Changed(ORSetKey("key")) =>
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Dead Code: Expression is Always false — Low

Package: test.scala.akka.cluster.ddata

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 616 (Dead Code: Expression is Always false) — Low

```
617 val _: ORSet[String] = c.dataValue
618 val elements3 = c.dataValue match {
619 case ORSet(elements3) => elements3
```

test/scala/akka/cluster/ddata/ORSetSpec.scala, line 630 (Dead Code: Expression is Always false) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/ORSetSpec.scala:630
Taint Flags:

```
627
628 val msg: Any = Changed(ORSetKey[String]("key"))(s1)
629 msg match {
630 case c @ Changed(ORSetKey("key")) =>
631 // FIXME we need to look into this for Scala 2.13.5
632 // val ORSet(elements3) = c.dataValue // `unapply(a: ReplicatedData)` is used here
633 // if `unapply(a: ReplicatedData)` isn't defined the next line doesn't compile:
```

test/scala/akka/cluster/ddata/PNCounterMapSpec.scala, line 84 (Dead Code: Expression is Always false) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/PNCounterMapSpec.scala:84
Taint Flags:

```
81 entries2 should be(Map("a" -> 1L, "b" -> 2L))
82
83 Changed(PNCounterMapKey[String]("key"))(m1) match {
84 case c @ Changed(PNCounterMapKey("key")) =>
85 val entries3 = c.dataValue match {
86 case PNCounterMap(entries3) => entries3
87 case _ => throw new RuntimeException()
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Dead Code: Expression is Always false — Low

Package: test.scala.akka.cluster.ddata

test/scala/akka/cluster/ddata/PNCounterSpec.scala, line 199 (Dead Code: Expression is Always false) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/PNCounterSpec.scala:199
Taint Flags:

```
196 value2 should be(1L)
197
198 Changed(PNCounterKey("key"))(c1) match {
199 case c @ Changed(PNCounterKey("key")) =>
200 val value3 = c.dataValue.value
201 val value4: BigInt = value3
202 value4 should be(1L)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Insecure Randomness (4 issues)

Abstract

Standard pseudorandom number generators cannot withstand cryptographic attacks.

Explanation

Insecure randomness errors occur when a function that can produce predictable values is used as a source of randomness in a security-sensitive context. Computers are deterministic machines, and as such are unable to produce true randomness. Pseudorandom Number Generators (PRNGs) approximate randomness algorithmically, starting with a seed from which subsequent values are calculated. There are two types of PRNGs: statistical and cryptographic. Statistical PRNGs provide useful statistical properties, but their output is highly predictable and form an easy to reproduce numeric stream that is unsuitable for use in cases where security depends on generated values being unpredictable. Cryptographic PRNGs address this problem by generating output that is more difficult to predict. For a value to be cryptographically secure, it must be impossible or highly improbable for an attacker to distinguish between the generated random value and a truly random value. In general, if a PRNG algorithm is not advertised as being cryptographically secure, then it is probably a statistical PRNG and should not be used in security-sensitive contexts, where its use can lead to serious vulnerabilities such as easy-to-guess temporary passwords, predictable cryptographic keys, session hijacking, and DNS spoofing. **Example:** The following code uses a statistical PRNG to create a URL for a receipt that remains active for some period of time after a purchase.

```
String GenerateReceiptURL(String baseUrl) {
    Random ranGen = new Random();
    ranGen.setSeed((new Date()).getTime());
    return (baseUrl + ranGen.nextInt(400000000) + ".html");
}
```

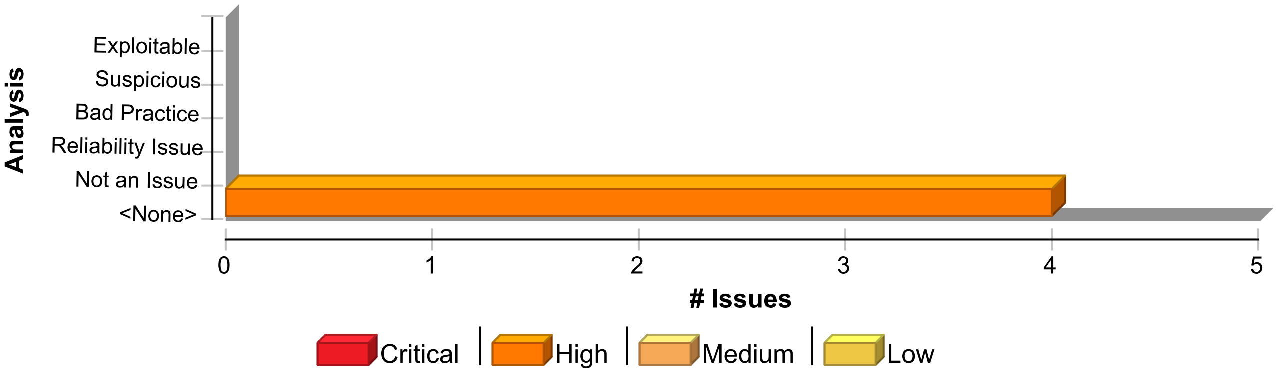
This code uses the `Random.nextInt()` function to generate "unique" identifiers for the receipt pages it generates. Since `Random.nextInt()` is a statistical PRNG, it is easy for an attacker to guess the strings it generates. Although the underlying design of the receipt system is also faulty, it would be more secure if it used a random number generator that did not produce predictable receipt identifiers, such as a cryptographic PRNG.

Recommendation

When unpredictability is critical, as is the case with most security-sensitive uses of randomness, use a cryptographic PRNG. Regardless of the PRNG you choose, always use a value with sufficient entropy to seed the algorithm. (Do not use values such as the current time because it offers only negligible entropy.) The Java language provides a cryptographic PRNG in `java.security.SecureRandom`. As is the case with other algorithm-based classes in `java.security`, `SecureRandom` provides an implementation-independent wrapper around a particular set of algorithms. When you request an instance of a `SecureRandom` object using `SecureRandom.getInstance()`, you can request a specific implementation of the algorithm. If the algorithm is available, then it is given as a `SecureRandom` object. If it is unavailable or if you do not specify a particular implementation, then you are given a `SecureRandom` implementation selected by the system. Sun provides a single `SecureRandom` implementation with the Java distribution named `SHA1PRNG`, which Sun describes as computing: "The SHA-1 hash over a true-random seed value concatenated with a 64-bit counter which is incremented by 1 for each operation. From the 160-bit SHA-1 output, only 64 bits are used [1]." However, the specifics of the Sun implementation of the `SHA1PRNG` algorithm are poorly documented, and it is unclear what sources of entropy the implementation uses and therefore what amount of true randomness exists in its output. Although there is speculation on the Web about the Sun implementation, there is no evidence to contradict the claim that the algorithm is cryptographically strong and can be used safely in security-sensitive contexts.

Issue Summary

© Copyright [2008-2022] Micro Focus or one of its affiliates.



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Insecure Randomness	4	0	0	4
Total	**4**	**0**	**0**	**4**

Insecure Randomness — High

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 2160 (Insecure Randomness) — High

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextInt()
Enclosing Method: selectRandomNode()
File: main/scala/akka/cluster/ddata/Replicator.scala:2160
Taint Flags:

```
2157  }
2158
2159  def selectRandomNode(addresses: immutable.IndexedSeq[UniqueAddress]): Option[UniqueAddress] =
2160  if (addresses.isEmpty) None else Some(addresses(ThreadLocalRandom.current.nextInt(addresses.size)))
2161
2162  def replica(node: UniqueAddress): ActorSelection =
2163  context.actorSelection(self.path.toStringWithAddress(node.address))
```

Package: main.scala.akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 2147 (Insecure Randomness) — High

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextInt()
Enclosing Method: apply()

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Insecure Randomness High

Package: main.scala.akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 2147 (Insecure Randomness) High

File: main/scala/akka/cluster/ddata/Replicator.scala:2147
Taint Flags:

```
2144 if (totChunks == statusTotChunks)
2145 statusCount += 1
2146 else {
2147 statusCount = ThreadLocalRandom.current.nextInt(0, totChunks)
2148 statusTotChunks = totChunks
2149 }
2150 val chunk = (statusCount % totChunks).toInt
```

Package: test.scala.akka.cluster.ddata

test/scala/akka/cluster/ddata/LotsOfDataBot.scala, line 106 (Insecure Randomness) High

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextBoolean()
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/LotsOfDataBot.scala:106
Taint Flags:

```
103 if (count <= 100)
104 replicator ! Subscribe(key, self)
105 val s = ThreadLocalRandom.current().nextInt(97, 123).toChar.toString
106 if (count <= maxEntries || ThreadLocalRandom.current().nextBoolean()) {
107 // add
108 replicator ! Update(key, ORSet(), WriteLocal)(_ :+ s)
109 } else {
```

test/scala/akka/cluster/ddata/LotsOfDataBot.scala, line 105 (Insecure Randomness) High

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextInt()
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/LotsOfDataBot.scala:105
Taint Flags:

```
102 val key = ORSetKey[String]((count % maxEntries).toString)
103 if (count <= 100)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Insecure Randomness — High

Package: test.scala.akka.cluster.ddata

test/scala/akka/cluster/ddata/LotsOfDataBot.scala, line 105 (Insecure Randomness) — High

```
104 replicator ! Subscribe(key, self)
105 val s = ThreadLocalRandom.current().nextInt(97, 123).toChar.toString
106 if (count <= maxEntries || ThreadLocalRandom.current().nextBoolean()) {
107 // add
108 replicator ! Update(key, ORSet(), WriteLocal)(_ :+ s)
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

J2EE Bad Practices: Leftover Debug Code (1 issue)

Abstract

Debug code can create unintended entry points in a deployed web application.

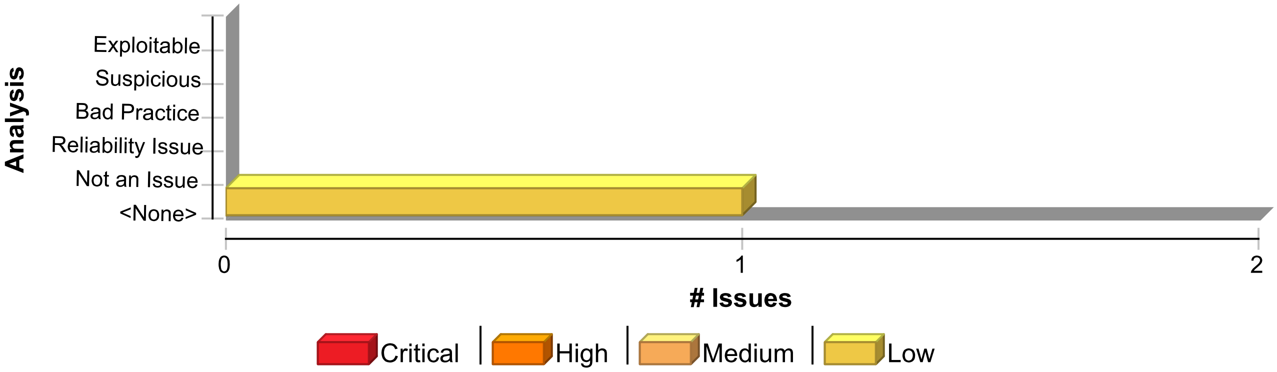
Explanation

A common development practice is to add "back door" code specifically designed for debugging or testing purposes that is not intended to be shipped or deployed with the application. When this sort of debug code is accidentally left in the application, the application is open to unintended modes of interaction. These back door entry points create security risks because they are not considered during design or testing and fall outside of the expected operating conditions of the application. The most common example of forgotten debug code is a `main()` method appearing in a web application. Although this is an acceptable practice during product development, classes that are part of a production J2EE application should not define a `main()`.

Recommendation

Remove debug code before deploying a production version of an application. Regardless of whether a direct security threat can be articulated, it is unlikely that there is a legitimate reason for such code to remain in the application after the early stages of development.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
J2EE Bad Practices: Leftover Debug Code	1	0	0	1
Total	**1**	**0**	**0**	**1**

J2EE Bad Practices: Leftover Debug Code	Low
Package: akka.cluster.ddata	
test/scala/akka/cluster/ddata/LotsOfDataBot.scala, line 24 (J2EE Bad Practices: Leftover Debug Code)	**Low**

Issue Details

Kingdom: Encapsulation
Scan Engine: SCA (Structural)

Sink Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

J2EE Bad Practices: Leftover Debug Code

Low

Package: akka.cluster.ddata

test/scala/akka/cluster/ddata/LotsOfDataBot.scala, line 24 (J2EE Bad Practices: Leftover Debug Code)

Low

Sink: Function: main
Enclosing Method: main()
File: test/scala/akka/cluster/ddata/LotsOfDataBot.scala:24
Taint Flags:

```
21  */
22 object LotsOfDataBot {
23
24  def main(args: Array[String]): Unit = {
25  if (args.isEmpty)
26  startup(Seq("2551", "2552", "0"))
27  else
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

J2EE Bad Practices: Threads (1 issue)

Abstract

Thread management in a web application is forbidden in some circumstances and is always highly error prone.

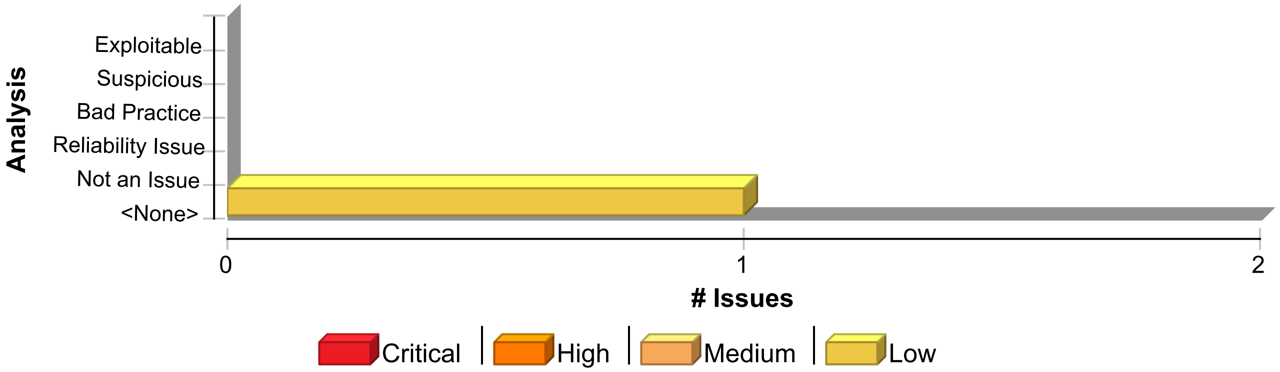
Explanation

Thread management in a web application is forbidden by the J2EE standard in some circumstances and is always highly error prone. Managing threads is difficult and is likely to interfere in unpredictable ways with the behavior of the application container. Even without interfering with the container, thread management usually leads to bugs that are hard to detect and diagnose like deadlock, race conditions, and other synchronization errors.

Recommendation

Avoid managing threads directly from within the web application. Instead use standards such as message driven beans and the EJB timer service that are provided by the application container.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
J2EE Bad Practices: Threads	1	0	0	1
Total	**1**	**0**	**0**	**1**

J2EE Bad Practices: Threads	**Low**
Package: test.scala.akka.cluster.ddata.protobuf	
test/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializerSpec.scala, line 278 (J2EE Bad Practices: Threads)	**Low**

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: sleep()
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializerSpec.scala:278

© Copyright [2008-2022] Micro Focus or one of its affiliates.

J2EE Bad Practices: Threads

Low

Package: test.scala.akka.cluster.ddata.protobuf

test/scala/akka/cluster/ddata/protobuf/ReplicatorMessageSerializerSpec.scala, line 278 (J2EE Bad Practices: Threads)

Low

Taint Flags:

```
275  cache.add(b, "B")
276  cache.add(c, "C")
277
278  Thread.sleep(30)
279  cache.evict()
280  cache.get(b) should be(null)
281  cache.get(c) should be(null)
```

Poor Style: Value Never Read (1 issue)

Abstract

The variable's value is assigned but never used, making it a dead store.

Explanation

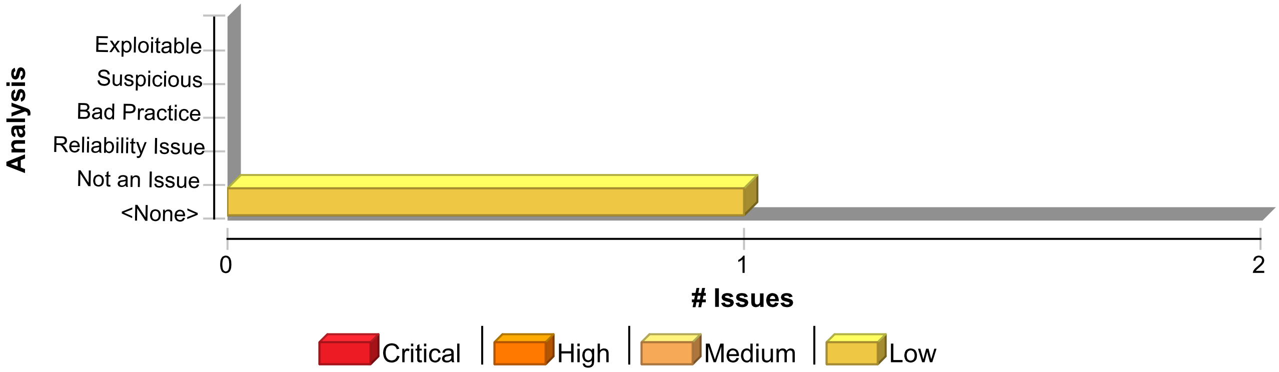
This variable's value is not used. After the assignment, the variable is either assigned another value or goes out of scope. **Example:** The following code excerpt assigns to the variable `r` and then overwrites the value without using it.

```
  r = getName();
  r = getNewBuffer(buf);
```

Recommendation

Remove unnecessary assignments in order to make the code easier to understand and maintain.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Poor Style: Value Never Read	1	0	0	1
Total	**1**	**0**	**0**	**1**

Poor Style: Value Never Read **Low**

Package: test.scala.akka.cluster.ddata.protobuf

test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala, line 162 (Poor Style: Value Never Read) **Low**

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: acc
Enclosing Method: apply()
File: test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala:162

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Poor Style: Value Never Read — Low

Package: test.scala.akka.cluster.ddata.protobuf

test/scala/akka/cluster/ddata/protobuf/ReplicatedDataSerializerSpec.scala, line 162 (Poor Style: Value Never Read) — Low

Taint Flags:

```
159
160 "serialize large ORSet" in {
161 val largeSet = (10000 until 20000).foldLeft(ORSet.empty[String]) {
162 case (acc, n) =>
163 val address = (n % 3) match {
164 case 0 => address1
165 case 1 => address2
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Unchecked Return Value (1 issue)

Abstract

Ignoring a method's return value can cause the program to overlook unexpected states and conditions.

Explanation

It is not uncommon for Java programmers to misunderstand `read()` and related methods that are part of many `java.io` classes. Most errors and unusual events in Java result in an exception being thrown. (This is one of the advantages that Java has over languages like C: Exceptions make it easier for programmers to think about what can go wrong.) But the stream and reader classes do not consider it unusual or exceptional if only a small amount of data becomes available. These classes simply add the small amount of data to the return buffer, and set the return value to the number of bytes or characters read. There is no guarantee that the amount of data returned is equal to the amount of data requested. This behavior makes it important for programmers to examine the return value from `read()` and other IO methods to ensure that they receive the amount of data they expect. **Example:** The following code loops through a set of users, reading a private data file for each user. The programmer assumes that the files are always exactly 1 kilobyte in size and therefore ignores the return value from `read()`. If an attacker can create a smaller file, the program will recycle the remainder of the data from the previous user and handle it as though it belongs to the attacker.

```
FileInputStream fis;
byte[] byteArray = new byte[1024];
for (Iterator i=users.iterator(); i.hasNext();) {
    String userName = (String) i.next();
    String pFileName = PFILE_ROOT + "/" + userName;
    FileInputStream fis = new FileInputStream(pFileName);
    fis.read(byteArray); // the file is always 1k bytes
    fis.close();
    processPFile(userName, byteArray);
}
```

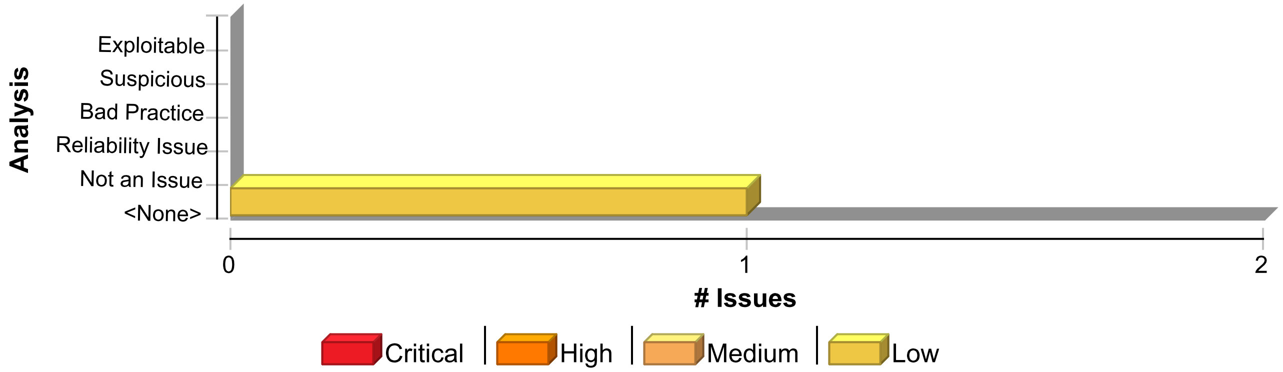
Recommendation

```
  FileInputStream fis;
  byte[] byteArray = new byte[1024];
  for (Iterator i=users.iterator(); i.hasNext();) {
    String userName = (String) i.next();
    String pFileName = PFILE_ROOT + "/" + userName;
    fis = new FileInputStream(pFileName);
    int bRead = 0;
    while (bRead < 1024) {
        int rd = fis.read(byteArray, bRead, 1024 - bRead);
        if (rd == -1) {
          throw new IOException("file is unusually small");
        }
        bRead += rd;
    }
    // could add check to see if file is too large here
    fis.close();
    processPFile(userName, byteArray);
  }
```

Note: Because the fix for this problem is relatively complicated, you might be tempted to use a simpler approach, such as checking the size of the file before you begin reading. Such an approach would render the application vulnerable to a file system race condition, whereby an attacker could replace a well-formed file with a malicious file between the file size check and the call to read data from the file.

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Unchecked Return Value	1	0	0	1
Total	**1**	**0**	**0**	**1**

Unchecked Return Value — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/DurableStore.scala, line 145 (Unchecked Return Value) — Low

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Semantic)

Sink Details

Sink: mkdirs()
Enclosing Method: akka$cluster$ddata$LmdbDurableStore$$lmdb()
File: main/scala/akka/cluster/ddata/DurableStore.scala:145
Taint Flags:

```
142 log.info("Using durable data in LMDB directory [{}]", dir.getCanonicalPath)
143 val env = {
144 val mapSize = config.getBytes("lmdb.map-size")
145 dir.mkdirs()
146 Env.create().setMapSize(mapSize).setMaxDbs(1).open(dir, EnvFlags.MDB_NOLOCK)
147 }
148
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Weak Cryptographic Hash (1 issue)

Abstract

Weak cryptographic hashes cannot guarantee data integrity and should not be used in security-critical contexts.

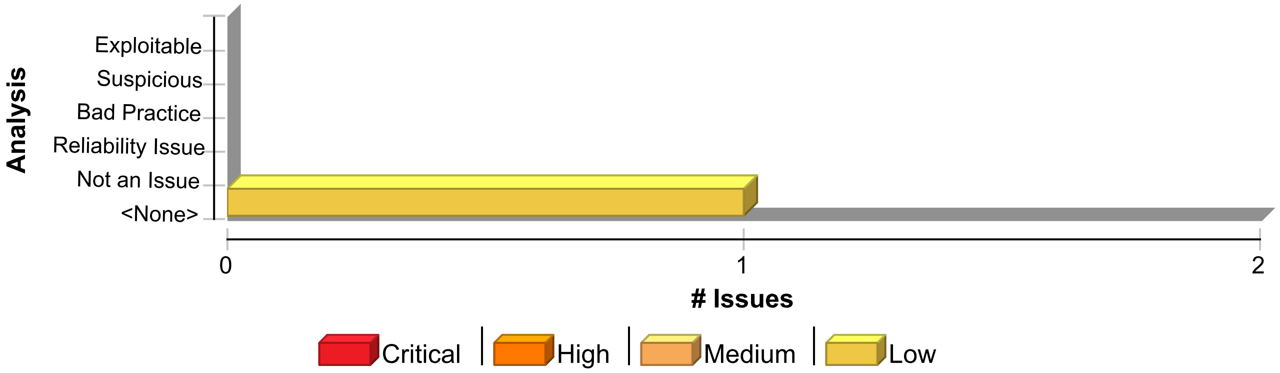
Explanation

MD2, MD4, MD5, RIPEMD-160, and SHA-1 are popular cryptographic hash algorithms often used to verify the integrity of messages and other data. However, as recent cryptanalysis research has revealed fundamental weaknesses in these algorithms, they should no longer be used within security-critical contexts. Effective techniques for breaking MD and RIPEMD hashes are widely available, so those algorithms should not be relied upon for security. In the case of SHA-1, current techniques still require a significant amount of computational power and are more difficult to implement. However, attackers have found the Achilles' heel for the algorithm, and techniques for breaking it will likely lead to the discovery of even faster attacks.

Recommendation

Discontinue the use of MD2, MD4, MD5, RIPEMD-160, and SHA-1 for data-verification in security-critical contexts. Currently, SHA-224, SHA-256, SHA-384, SHA-512, and SHA-3 are good alternatives. However, these variants of the Secure Hash Algorithm have not been scrutinized as closely as SHA-1, so be mindful of future research that might impact the security of these algorithms.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Weak Cryptographic Hash	1	0	0	1
Total	**1**	**0**	**0**	**1**

Weak Cryptographic Hash

Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 2000 (Weak Cryptographic Hash)

Low

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

© Copyright [2008-2022] Micro Focus or one of its affiliates.

Weak Cryptographic Hash — Low

Package: akka.cluster.ddata

main/scala/akka/cluster/ddata/Replicator.scala, line 2000 (Weak Cryptographic Hash) — Low

Sink: getInstance()
Enclosing Method: digest()
File: main/scala/akka/cluster/ddata/Replicator.scala:2000
Taint Flags:

```
1997  if (envelope.data == DeletedData) (DeletedDigest, 0)
1998  else {
1999  val bytes = serializer.toBinary(envelope.withoutDeltaVersions)
2000  val dig = ByteString.fromArray(MessageDigest.getInstance("SHA-1").digest(bytes))
2001  (dig, bytes.length)
2002  }
2003
```

© Copyright [2008-2022] Micro Focus or one of its affiliates.

© Copyright [2008-2022] Micro Focus or one of its affiliates.