



Fortify Standalone Report Generator

Developer Workbook

akka-cluster



Table of Contents

- [Executive Summary](#)
- [Project Description](#)
- [Issue Breakdown by Fortify Categories](#)
- [Results Outline](#)

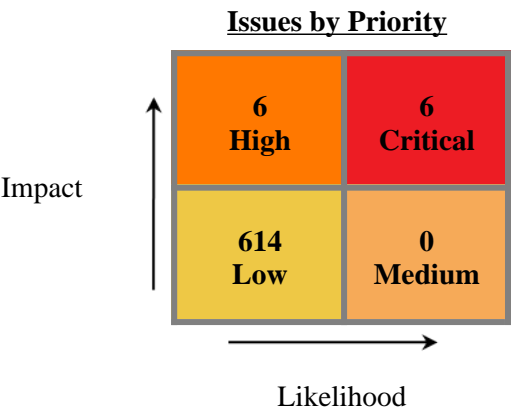


Executive Summary

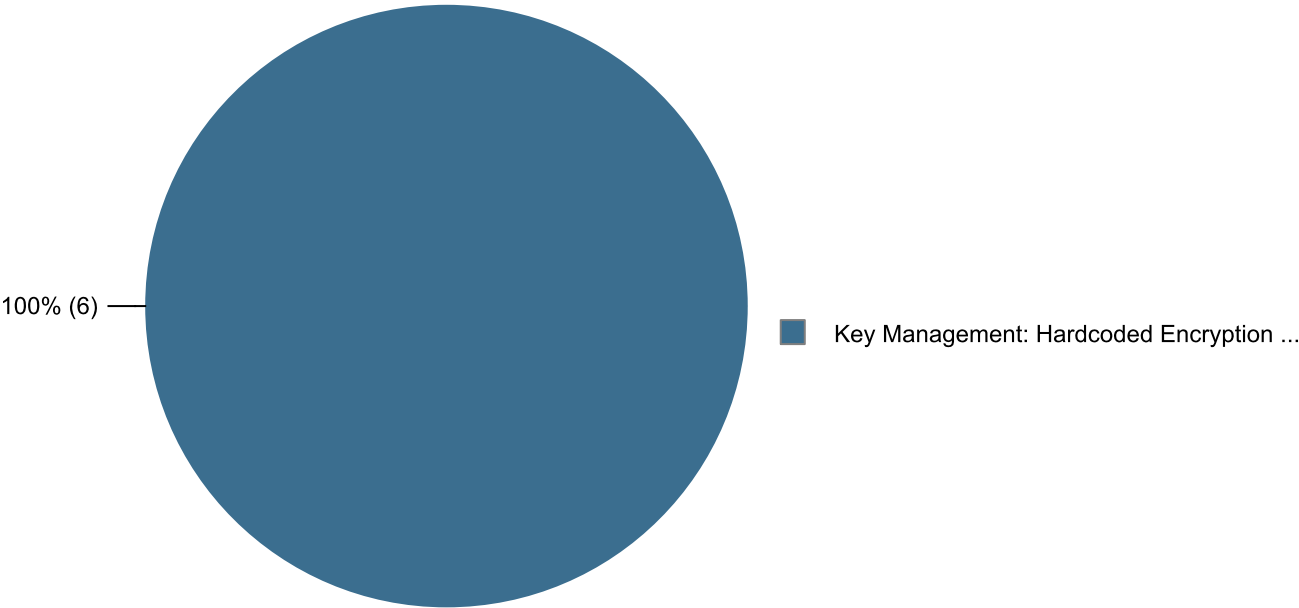
This workbook is intended to provide all necessary details and information for a developer to understand and remediate the different issues discovered during the akka-cluster project audit. The information contained in this workbook is targeted at project managers and developers.

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

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



Top Ten Critical Categories



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, 11:19 AM	Engine Version:	21.1.1.0009
Host Name:	Jacks-Work-MBP.local	Certification:	VALID
Number of Files:	67	Lines of Code:	8,194

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



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: Class Does Not Implement equals	0	0	0	0 / 221	0 / 221
Code Correctness: Constructor Invokes Overridable Function	0	0	0	0 / 337	0 / 337
Code Correctness: Erroneous String Compare	0	0	0	0 / 19	0 / 19
Code Correctness: Non-Static Inner Class Implements Serializable	0	0	0	0 / 19	0 / 19
Dead Code: Expression is Always false	0	0	0	0 / 1	0 / 1
Insecure Randomness	0	0 / 6	0	0	0 / 6
J2EE Bad Practices: Threads	0	0	0	0 / 13	0 / 13
Key Management: Hardcoded Encryption Key	0 / 6	0	0	0	0 / 6
Poor Style: Value Never Read	0	0	0	0 / 2	0 / 2
Weak Cryptographic Hash	0	0	0	0 / 2	0 / 2



Results Outline

Code Correctness: Class Does Not Implement equals (221 issues)

Abstract

The `equals()` method is called on an object that does not implement `equals()`.

Explanation

When comparing objects, developers usually want to compare properties of objects. However, calling `equals()` on a class (or any super class/interface) that does not explicitly implement `equals()` results in a call to the `equals()` method inherited from `java.lang.Object`. Instead of comparing object member fields or other properties, `Object.equals()` compares two object instances to see if they are the same. Although there are legitimate uses of `Object.equals()`, it is often an indication of buggy code. **Example 1:**

```
public class AccountGroup
{
    private int gid;

    public int getGid()
    {
        return gid;
    }

    public void setGid(int newGid)
    {
        gid = newGid;
    }
}
...
public class CompareGroup
{
    public boolean compareGroups(AccountGroup group1, AccountGroup group2)
    {
        return group1.equals(group2);    //equals() is not implemented in
AccountGroup
    }
}
```

Recommendation

Verify that the use of `Object.equals()` is really the method you intend to call. If not, implement an `equals()` method or use a different method for comparing objects. **Example 2:** The following code adds an `equals()` method to the example from the Explanation section.

```
public class AccountGroup
{
    private int gid;

    public int getGid()
    {
        return gid;
    }

    public void setGid(int newGid)
```



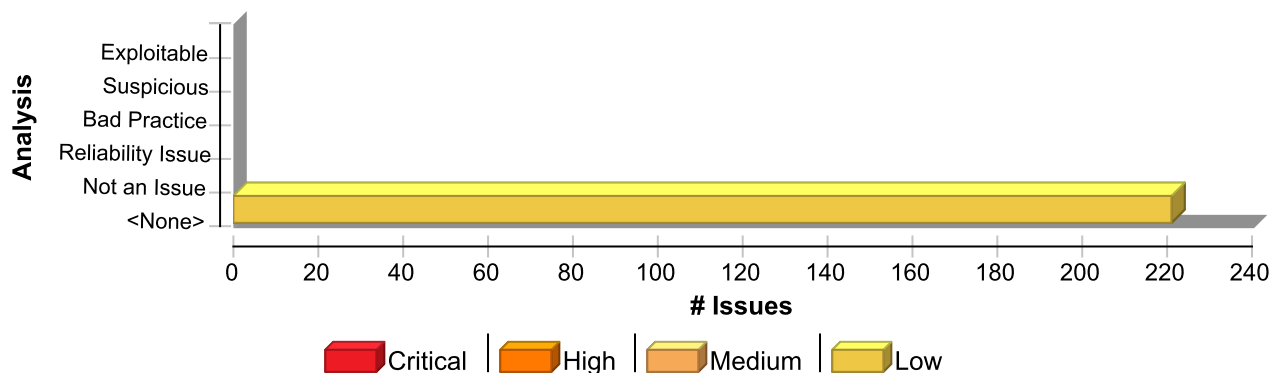
```

    {
        gid = newGid;
    }

    public boolean equals(Object o)
    {
        if (!(o instanceof AccountGroup))
            return false;
        AccountGroup other = (AccountGroup) o;
        return (gid == other.getGid());
    }
}
...
public class CompareGroup
{
    public static boolean compareGroups(AccountGroup group1, AccountGroup
group2)
    {
        return group1.equals(group2);
    }
}

```

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Class Does Not Implement equals	221	0	0	221
Total	221	0	0	221

Code Correctness: Class Does Not Implement equals

Low

Package: akka.cluster

main/scala/akka/cluster/MembershipState.scala, line 153 (Code Correctness: Class Does Not Implement equals)

Low

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/MembershipState.scala, line 153 (Code Correctness: Class Does Not Implement equals)	Low

Sink: FunctionCall: equals
Enclosing Method: isReachableExcludingDownedObservers()
File: main/scala/akka/cluster/MembershipState.scala:153
Taint Flags:

```

150 val to = latestGossip.member(toAddress)
151
152 // if member is in the same data center, we ignore cross data center unreachability
153 if (selfDc == to.dataCenter) dcReachabilityExcludingDownedObservers.isReachable(toAddress)
154 // if not it is enough that any non-downed node observed it as unreachable
155 else latestGossip.reachabilityExcludingDownedObservers.isReachable(toAddress)
156 }

```

main/scala/akka/cluster/ClusterEvent.scala, line 563 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: applyOrElse()
File: main/scala/akka/cluster/ClusterEvent.scala:563
Taint Flags:

```

560 val membersGroupedByAddress = List(newGossip.members, oldGossip.members).flatten.groupBy(_.uniqueAddress)
561 val changedMembers = membersGroupedByAddress.collect {
562   case (_, newMember :: oldMember :: Nil)
563     if newMember.status != oldMember.status || newMember.upNumber != oldMember.upNumber =>
564     newMember
565 }
566 val memberEvents = (newMembers ++ changedMembers).unsorted.collect {

```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 302 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: activeReceiversIn()
File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:302
Taint Flags:



Code Correctness: Class Does Not Implement equals**Low****Package: akka.cluster****main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 302 (Code Correctness: Class Does Not Implement equals)****Low**

```
299
300 /** Lists addresses in given DataCenter that this node should send heartbeats to */
301 private def activeReceiversIn(dc: DataCenter): Set[UniqueAddress] =
302 if (dc == selfDataCenter) Set.empty // CrossDcHeartbeatSender is not supposed to send within its own Dc
303 else {
304 val otherNodes = state.getOrElse(dc, emptyMembersSortedSet)
305 otherNodes.take(nrOfMonitoredNodesPerDc).iterator.map(_.uniqueAddress).to(immutable.Set)
```

main/scala/akka/cluster/MembershipState.scala, line 303 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: applyOrElse()
File: main/scala/akka/cluster/MembershipState.scala:303
Taint Flags:

```
300 latestGossip.members.iterator
301 .collect {
302 case m
303 if m.dataCenter == state.selfDc && !latestGossip.seenByNode(m.uniqueAddress) && state
304 .validNodeForGossip(m.uniqueAddress) =>
305 m.uniqueAddress
306 }
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1347 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: akka\$cluster\$ClusterCoreDaemon\$\$isJoiningToUp()
File: main/scala/akka/cluster/ClusterDaemon.scala:1347
Taint Flags:

```
1344
1345 val changedMembers = {
1346 val enoughMembers: Boolean = isMinNrOfMembersFulfilled
```



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 1347 (Code Correctness: Class Does Not Implement equals)****Low**

```
1347 def isJoiningToUp(m: Member): Boolean = (m.status == Joining || m.status == WeaklyUp) && enoughMembers
1348
1349 latestGossip.members.collect {
1350   var upNumber = 0
```

main/scala/akka/cluster/MembershipState.scala, line 283 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()**File:** main/scala/akka/cluster/MembershipState.scala:283**Taint Flags:**

```
280 } else {
281 // single dc or not among the N oldest - select local nodes
282 val selectedNodes = state.members.toVector.collect {
283 case m if m.dataCenter == state.selfDc && state.validNodeForGossip(m.uniqueAddress) => m.uniqueAddress
284 }
285
286 if (selectedNodes.size <= n) selectedNodes
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1371 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1371**Taint Flags:**

```
1368 // Move LEAVING => EXITING (once we have a convergence on LEAVING)
1369 m.copy(status = Exiting)
1370
1371 case m if m.dataCenter == selfDc & m.status == PreparingForShutdown =>
1372 // Move PreparingForShutdown => ReadyForShutdown (once we have a convergence on PreparingForShutdown)
1373 m.copy(status = ReadyForShutdown)
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 1371 (Code Correctness: Class Does Not Implement equals)	Low

```
1374 }
```

main/scala/akka/cluster/ClusterDaemon.scala, line 743 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: joining()
File: main/scala/akka/cluster/ClusterDaemon.scala:743
Taint Flags:

```
740 def joining(joiningNode: UniqueAddress, roles: Set[String], appVersion: Version): Unit = {
741   if (!preparingForShutdown) {
742     val selfStatus = latestGossip.member(selfUniqueAddress).status
743     if (joiningNode.address.protocol != selfAddress.protocol)
744       logWarning(
745         "Member with wrong protocol tried to join, but was ignored, expected [{}] but was [{})",
746         selfAddress.protocol,
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1289 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: applyOrElse()
File: main/scala/akka/cluster/ClusterDaemon.scala:1289
Taint Flags:

```
1286 // When all reachable have seen the state this member will shutdown itself when it has
1287 // status Down. The down commands should spread before we shutdown.
1288 val unreachable = membershipState.dcReachability.allUnreachableOrTerminated
1289 val downed = membershipState.dcMembers.collect { case m if m.status == Down => m.uniqueAddress }
1290 if (selfDownCounter >= MaxTicksBeforeShuttingDownMyself || downed.forall(
1291   node => unreachable(node) || latestGossip.seenByNode(node))) {
1292 // the reason for not shutting down immediately is to give the gossip a chance to spread
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterEvent.scala, line 311 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: ClusterEvent\$MemberUp() File: main/scala/akka/cluster/ClusterEvent.scala:311 Taint Flags:	
<pre> 308 * Member status changed to Up. 309 */ 310 final case class MemberUp(member: Member) extends MemberEvent { 311 if (member.status != Up) throw new IllegalArgumentException("Expected Up status, got: " + member) 312 } 313 314 /** </pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 963 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: downing() File: main/scala/akka/cluster/ClusterDaemon.scala:963 Taint Flags:	
<pre> 960 961 // check if the node to DOWN is in the `members` set 962 localMembers.find(_.address == address) match { 963 case Some(m) if m.status != Down => 964 if (localReachability.isReachable(m.uniqueAddress)) 965 logInfo(966 ClusterLogMarker.memberChanged(m.uniqueAddress, MemberStatus.Down), </pre>	
main/scala/akka/cluster/ClusterReadView.scala, line 202 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterReadView.scala, line 202 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** isAvailable()**File:** main/scala/akka/cluster/ClusterReadView.scala:202**Taint Flags:**

```
199 def isAvailable: Boolean = {  
200   val myself = self  
201   !unreachableMembers.contains(myself) &&  
202   myself.status != MemberStatus.Down &&  
203   myself.status != MemberStatus.Removed  
204 }  
205
```

main/scala/akka/cluster/ClusterEvent.scala, line 570 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()**File:** main/scala/akka/cluster/ClusterEvent.scala:570**Taint Flags:**

```
567 case m if m.status == Joining => MemberJoined(m)  
568 case m if m.status == WeaklyUp => MemberWeaklyUp(m)  
569 case m if m.status == Up => MemberUp(m)  
570 case m if m.status == Leaving => MemberLeft(m)  
571 case m if m.status == Exiting => MemberExited(m)  
572 case m if m.status == Down => MemberDowned(m)  
573 case m if m.status == PreparingForShutdown => MemberPreparingForShutdown(m)
```

main/scala/akka/cluster/ClusterEvent.scala, line 344 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** ClusterEvent\$MemberDowned()

Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterEvent.scala, line 344 (Code Correctness: Class Does Not Implement equals)	Low

File: main/scala/akka/cluster/ClusterEvent.scala:344

Taint Flags:

```

341 * when all members have seen the `Down` status.
342 */
343 final case class MemberDowned(member: Member) extends MemberEvent {
344 if (member.status != Down) throw new IllegalArgumentException("Expected Down status, got: " + member)
345 }
346
347 /**

```

main/scala/akka/cluster/JoinConfigCompatChecker.scala, line 75 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: checkCompat()

File: main/scala/akka/cluster/JoinConfigCompatChecker.scala:75

Taint Flags:

```

72 actualConfig: Config): ConfigValidation = {
73
74 def checkCompat(key: String, value: ConfigValue) = {
75 actualConfig.hasPath(key) && actualConfig.getValue(key) == value
76 }
77
78 // retrieve all incompatible keys

```

main/scala/akka/cluster/ClusterReadView.scala, line 203 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: isAvailable()

File: main/scala/akka/cluster/ClusterReadView.scala:203

Taint Flags:

```

200 val myself = self

```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterReadView.scala, line 203 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 201 !unreachableMembers.contains(myself) && 202 myself.status != MemberStatus.Down && 203 myself.status != MemberStatus.Removed 204 } 205 206 def reachability: Reachability = _state.get().reachability </pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 1262 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: applyOrElse() File: main/scala/akka/cluster/ClusterDaemon.scala:1262 Taint Flags:	
<pre> 1259 membershipState.dcReachabilityExcludingDownedObservers, 1260 latestGossip.members 1261 .collect { 1262 case m if m.dataCenter == selfDc => 1263 s"\${m.address} {m.status} seen=\${latestGossip.seenByNode(m.uniqueAddress)}" 1264 } 1265 .mkString(", ") </pre>	
main/scala/akka/cluster/ClusterEvent.scala, line 294 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: ClusterEvent\$MemberJoined() File: main/scala/akka/cluster/ClusterEvent.scala:294 Taint Flags:	
<pre> 291 * Member status changed to Joining. 292 */ 293 final case class MemberJoined(member: Member) extends MemberEvent { 294 if (member.status != Joining) throw new IllegalArgumentException("Expected Joining status, got: " + member) </pre>	



Code Correctness: Class Does Not Implement equals	Low
--	------------

Package: akka.cluster

main/scala/akka/cluster/ClusterEvent.scala, line 294 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

```
295 }
296
297 /**
```

main/scala/akka/cluster/Member.scala, line 342 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: equals()
File: main/scala/akka/cluster/Member.scala:342
Taint Flags:

```
339
340 override def equals(obj: Any): Boolean =
341   obj match {
342     case ua: UniqueAddress => this.address.equals(ua.address) && this.longUid.equals(ua.longUid)
343     case _ => false
344   }
345
```

test/scala/akka/cluster/FailureDetectorPuppet.scala, line 33 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: isAvailable()
File: test/scala/akka/cluster/FailureDetectorPuppet.scala:33
Taint Flags:

```
30
31 override def isAvailable: Boolean = status.get match {
32   case Unknown | Up => true
33   case Down => false
34
35 }
36
```



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterRemoteWatcher.scala, line 194 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** isWatchOutsideClusterAllowed()**File:** main/scala/akka/cluster/ClusterRemoteWatcher.scala:194**Taint Flags:**

```
191 * recovered state.  
192 */  
193 private def isWatchOutsideClusterAllowed(watchee: InternalActorRef): Boolean = {  
194   context.system.name == watchee.path.address.system && {  
195     val pathPrefix = watchee.path.elements.take(2).mkString("/", "/", "/")  
196     watchPathAllowList.contains(pathPrefix)  
197   }
```

main/scala/akka/cluster/Cluster.scala, line 581 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** logError()**File:** main/scala/akka/cluster/Cluster.scala:581**Taint Flags:**

```
578 logAtLevel(Logging.ErrorLevel, log.format(template, arg1, arg2, arg3))  
579  
580 def logError(cause: Throwable, message: String): Unit = {  
581   if (settings.SelfDataCenter == ClusterSettings.DefaultDataCenter)  
582     log.error(cause, "Cluster Node [{}] - {}", selfAddress, message)  
583   else  
584     log.error(cause, "Cluster Node [{}] dc [{}] - {}", selfAddress, settings.SelfDataCenter, message)
```

test/scala/akka/cluster/FailureDetectorPuppet.scala, line 32 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**test/scala/akka/cluster/FailureDetectorPuppet.scala, line 32 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** isAvailable()**File:** test/scala/akka/cluster/FailureDetectorPuppet.scala:32**Taint Flags:**

29 def markNodeAsAvailable(): Unit = status.set(Up)

30

31 override def isAvailable: Boolean = status.get match {

32 case Unknown | Up => true

33 case Down => false

34

35 }

main/scala/akka/cluster/ClusterDaemon.scala, line 899 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** exitingCompleted()**File:** main/scala/akka/cluster/ClusterDaemon.scala:899**Taint Flags:**

896 // ExitingCompleted sent via CoordinatedShutdown to continue the leaving process.

897 exitingTasksInProgress = false

898 // status Removed also before joining

899 if (membershipState.selfMember.status != MemberStatus.Removed) {

900 // mark as seen

901 membershipState = membershipState.seen()

902 assertLatestGossip()

main/scala/akka/cluster/ClusterActorRefProvider.scala, line 101 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** parseConfig()

Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterActorRefProvider.scala, line 101 (Code Correctness: Class Does Not Implement equals)	Low

File: main/scala/akka/cluster/ClusterActorRefProvider.scala:101

Taint Flags:

```

98 super.parseConfig(path, config2) match {
99 case d @ Some(deploy) =>
100 if (deploy.config.getBoolean("cluster.enabled")) {
101 if (deploy.scope != NoScopeGiven)
102 throw new ConfigurationException(
103 "Cluster deployment can't be combined with scope [%s]".format(deploy.scope))
104 if (deploy.routerConfig.isInstanceOf[RemoteRouterConfig])

```

main/scala/akka/cluster/ClusterEvent.scala, line 569 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: applyOrElse()

File: main/scala/akka/cluster/ClusterEvent.scala:569

Taint Flags:

```

566 val memberEvents = (newMembers ++ changedMembers).unsorted.collect {
567 case m if m.status == Joining => MemberJoined(m)
568 case m if m.status == WeaklyUp => MemberWeaklyUp(m)
569 case m if m.status == Up => MemberUp(m)
570 case m if m.status == Leaving => MemberLeft(m)
571 case m if m.status == Exiting => MemberExited(m)
572 case m if m.status == Down => MemberDowned(m)

```

main/scala/akka/cluster/MembershipState.scala, line 191 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: isInSameDc()

File: main/scala/akka/cluster/MembershipState.scala:191

Taint Flags:

```

188 }

```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/MembershipState.scala, line 191 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 189 190 def isInSameDc(node: UniqueAddress): Boolean = 191 node == selfUniqueAddress latestGossip.member(node).dataCenter == selfDc 192 193 /** 194 * Never gossip to self and not to node marked as unreachable by self (heartbeat </pre>	
main/scala/akka/cluster/Reachability.scala, line 280 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: applyOrElse() File: main/scala/akka/cluster/Reachability.scala:280 Taint Flags:	
<pre> 277 records.groupBy(_._subject).collect { 278 case (subject, records) if records.exists(_._status == Unreachable) => 279 val observers: Set[UniqueAddress] = 280 records.iterator.collect { case r if r._status == Unreachable => r._observer }.to(immutable.Set) 281 subject -> observers 282 } 283 } </pre>	
main/scala/akka/cluster/ClusterEvent.scala, line 327 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: ClusterEvent\$MemberReadyForShutdown() File: main/scala/akka/cluster/ClusterEvent.scala:327 Taint Flags:	
<pre> 324 } 325 326 final case class MemberReadyForShutdown(member: Member) extends MemberEvent { 327 if (member._status != ReadyForShutdown) </pre>	



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterEvent.scala, line 327 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 328 throw new IllegalArgumentException("Expected ReadyForShutdown status, got: " + member) 329 } 330 </pre>	
main/scala/akka/cluster/MembershipState.scala, line 96 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: applyOrElse() File: main/scala/akka/cluster/MembershipState.scala:96 Taint Flags:	
<pre> 93 * nodes outside of the data center 94 */ 95 lazy val dcReachability: Reachability = 96 overview.reachability.removeObservers(members.collect { case m if m.dataCenter != selfDc => m.uniqueAddress }) 97 98 /** 99 * @return Reachability excluding observations from nodes outside of the data center and observations within self data center, </pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 1390 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: leaderActionsOnConvergence() File: main/scala/akka/cluster/ClusterDaemon.scala:1390 Taint Flags:	
<pre> 1387 val newGossip = 1388 latestGossip.update(changedMembers).removeAll(removed, System.currentTimeMillis()) 1389 1390 if (!exitingTasksInProgress && newGossip.member(selfUniqueAddress).status == Exiting) { 1391 // Leader is moving itself from Leaving to Exiting. 1392 // ExitingCompleted will be received via CoordinatedShutdown to continue 1393 // the leaving process. Meanwhile the gossip state is not marked as seen. </pre>	



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 1168 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** receiveGossip()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1168**Taint Flags:**

```
1165 coordShutdown.run(CoordinatedShutdown.ClusterLeavingReason)
1166 }
1167
1168 if (selfStatus == Down && localGossip.member(selfUniqueAddress).status != Down) {
1169   logWarning("Received gossip where this member has been downed, from [{ }]", from.address)
1170   shutdownSelfWhenDown()
1171 }
```

main/scala/akka/cluster/ClusterDaemon.scala, line 699 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** join()**File:** main/scala/akka/cluster/ClusterDaemon.scala:699**Taint Flags:**

```
696 "Trying to join member with wrong protocol, but was ignored, expected [{ }] but was [{ }]",
697 selfAddress.protocol,
698 address.protocol)
699 else if (address.system != selfAddress.system)
700 logWarning(
701 "Trying to join member with wrong ActorSystem name, but was ignored, expected [{ }] but was [{ }]",
702 selfAddress.system,
```

main/scala/akka/cluster/ClusterSettings.scala, line 149 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 149 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:149**Taint Flags:**

```
146 }  
147 }  
148  
149 val AllowWeaklyUpMembers: Boolean = WeaklyUpAfter != Duration.Zero  
150  
151 val SelfDataCenter: DataCenter = cc.getString("multi-data-center.self-data-center")  
152
```

main/scala/akka/cluster/Reachability.scala, line 271 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()**File:** main/scala/akka/cluster/Reachability.scala:271**Taint Flags:**

```
268 case Some(observerRows) =>  
269 observerRows.iterator  
270 .collect {  
271 case (subject, record) if record.status == Unreachable => subject  
272 }  
273 .to(immutable.Set)  
274 }
```

main/scala/akka/cluster/ClusterEvent.scala, line 503 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterEvent.scala, line 503 (Code Correctness: Class Does Not Implement equals)****Low****File:** main/scala/akka/cluster/ClusterEvent.scala:503**Taint Flags:**

```
500 @InternalApi
501 private[cluster] def isDataCenterReachable(state: MembershipState)(otherDc: DataCenter): Boolean = {
502   val unrelatedDcNodes = state.latestGossip.members.collect {
503     case m if m.dataCenter != otherDc && m.dataCenter != state.selfDc => m.uniqueAddress
504   }
505
506   val reachabilityForOtherDc = state.dcReachabilityWithoutObservationsWithin.remove(unrelatedDcNodes)
```

main/scala/akka/cluster/ClusterEvent.scala, line 503 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()**File:** main/scala/akka/cluster/ClusterEvent.scala:503**Taint Flags:**

```
500 @InternalApi
501 private[cluster] def isDataCenterReachable(state: MembershipState)(otherDc: DataCenter): Boolean = {
502   val unrelatedDcNodes = state.latestGossip.members.collect {
503     case m if m.dataCenter != otherDc && m.dataCenter != state.selfDc => m.uniqueAddress
504   }
505
506   val reachabilityForOtherDc = state.dcReachabilityWithoutObservationsWithin.remove(unrelatedDcNodes)
```

main/scala/akka/cluster/ClusterEvent.scala, line 318 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** ClusterEvent\$MemberLeft()**File:** main/scala/akka/cluster/ClusterEvent.scala:318**Taint Flags:**

```
315 * Member status changed to Leaving.
```



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterEvent.scala, line 318 (Code Correctness: Class Does Not Implement equals)****Low**

```
316 */
317 final case class MemberLeft(member: Member) extends MemberEvent {
318   if (member.status != Leaving) throw new IllegalArgumentException("Expected Leaving status, got: " + member)
319 }
320
321 final case class MemberPreparingForShutdown(member: Member) extends MemberEvent {
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1648 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** postStop()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1648**Taint Flags:**

```
1645 cluster.subscribe(self, to)
1646
1647 override def postStop(): Unit = {
1648   if (status == Removed)
1649     done()
1650   cluster.unsubscribe(self)
1651 }
```

main/scala/akka/cluster/Member.scala, line 45 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** toString()**File:** main/scala/akka/cluster/Member.scala:45**Taint Flags:**

```
42 case _ => false
43 }
44 override def toString: String = {
45   s"Member($address, $status${if (dataCenter == ClusterSettings.DefaultDataCenter) "" else s", $dataCenter"})${if (appVersion == Version.Zero) ""
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/Member.scala, line 45 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 46 else s", \$appVersion"}}" 47 } 48 </pre>	
main/scala/akka/cluster/MembershipState.scala, line 114 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: applyOrElse() File: main/scala/akka/cluster/MembershipState.scala:114 Taint Flags:	
<pre> 111 val membersToExclude = members.collect { case m if m.status == Down m.dataCenter != selfDc => m.uniqueAddress } 112 overview.reachability 113 .removeObservers(membersToExclude) 114 .remove(members.collect { case m if m.dataCenter != selfDc => m.uniqueAddress }) 115 } 116 117 lazy val dcReachabilityNoOutsideNodes: Reachability = </pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 1353 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: applyOrElse() File: main/scala/akka/cluster/ClusterDaemon.scala:1353 Taint Flags:	
<pre> 1350 var upNumber = 0 1351 1352 { 1353 case m if m.dataCenter == selfDc && isJoiningToUp(m) && !preparingForShutdown => 1354 // Move JOINING => UP (once all nodes have seen that this node is JOINING, i.e. we have a convergence) 1355 // and minimum number of nodes have joined the cluster 1356 // don't move members to up when preparing for shutdown </pre>	



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 1367 (Code Correctness: Class Does Not Implement equals)	Low

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: applyOrElse()
File: main/scala/akka/cluster/ClusterDaemon.scala:1367
Taint Flags:

```

1364 }
1365 m.copyUp(upNumber)
1366
1367 case m if m.dataCenter == selfDc && m.status == Leaving =>
1368 // Move LEAVING => EXITING (once we have a convergence on LEAVING)
1369 m.copy(status = Exiting)
1370

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1371 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: applyOrElse()
File: main/scala/akka/cluster/ClusterDaemon.scala:1371
Taint Flags:

```

1368 // Move LEAVING => EXITING (once we have a convergence on LEAVING)
1369 m.copy(status = Exiting)
1370
1371 case m if m.dataCenter == selfDc & m.status == PreparingForShutdown =>
1372 // Move PreparingForShutdown => ReadyForShutdown (once we have a convergence on PreparingForShutdown)
1373 m.copy(status = ReadyForShutdown)
1374 }

```

test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala, line 31 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala, line 31 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** isAvailable()**File:** test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala:31**Taint Flags:**

```
28 def markNodeAsAvailable(): Unit = status = Up
29
30 override def isAvailable: Boolean = status match {
31   case Unknown | Up => true
32   case Down => false
33 }
34
```

main/scala/akka/cluster/ClusterEvent.scala, line 322 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** ClusterEvent\$MemberPreparingForShutdown()**File:** main/scala/akka/cluster/ClusterEvent.scala:322**Taint Flags:**

```
319 }
320
321 final case class MemberPreparingForShutdown(member: Member) extends MemberEvent {
322   if (member.status != PreparingForShutdown)
323     throw new IllegalArgumentException("Expected PreparingForShutdown status, got: " + member)
324 }
325
```

main/scala/akka/cluster/MembershipState.scala, line 257 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/MembershipState.scala, line 257 (Code Correctness: Class Does Not Implement equals)****Low****File:** main/scala/akka/cluster/MembershipState.scala:257**Taint Flags:**

```
254 if (state.latestGossip.isMultiDc && state.ageSortedTopOldestMembersPerDc(state.selfDc).contains(state.selfMember)) {  
255 // this node is one of the N oldest in the cluster, gossip to one cross-dc but mostly locally  
256 val randomLocalNodes = Random.shuffle(state.members.toVector.collect {  
257 case m if m.dataCenter == state.selfDc && state.validNodeForGossip(m.uniqueAddress) => m.uniqueAddress  
258 })  
259  
260 @tailrec
```

main/scala/akka/cluster/Reachability.scala, line 146 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** change()**File:** main/scala/akka/cluster/Reachability.scala:146**Taint Flags:**

```
143 } else  
144 new Reachability(records :+ newRecord, newVersions)  
145 case Some(oldRecord) =>  
146 if (oldRecord.status == Terminated || oldRecord.status == status)  
147 this  
148 else {  
149 if (status == Reachable && oldObserverRows.forall {
```

main/scala/akka/cluster/MembershipState.scala, line 111 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()**File:** main/scala/akka/cluster/MembershipState.scala:111**Taint Flags:**

```
108 * @return reachability for data center nodes, with observations from outside the data center or from downed nodes filtered out
```



Code Correctness: Class Does Not Implement equals**Low****Package: akka.cluster****main/scala/akka/cluster/MembershipState.scala, line 111 (Code Correctness: Class Does Not Implement equals)****Low**

```
109 */
110 lazy val dcReachabilityExcludingDownedObservers: Reachability = {
111 val membersToExclude = members.collect { case m if m.status == Down || m.dataCenter != selfDc => m.uniqueAddress }
112 overview.reachability
113 .removeObservers(membersToExclude)
114 .remove(members.collect { case m if m.dataCenter != selfDc => m.uniqueAddress })
```

main/scala/akka/cluster/ClusterEvent.scala, line 355 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** ClusterEvent\$MemberRemoved()**File:** main/scala/akka/cluster/ClusterEvent.scala:355**Taint Flags:**

```
352 * after graceful leaving and exiting.
353 */
354 final case class MemberRemoved(member: Member, previousStatus: MemberStatus) extends MemberEvent {
355 if (member.status != Removed) throw new IllegalArgumentException("Expected Removed status, got: " + member)
356 }
357
358 /**
```

main/scala/akka/cluster/ClusterDaemon.scala, line 415 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** ClusterCoreDaemon()**File:** main/scala/akka/cluster/ClusterDaemon.scala:415**Taint Flags:**

```
412
413 // start periodic publish of current stats
414 val publishStatsTask: Option[Cancellable] = PublishStatsInterval match {
415 case Duration.Zero | _: Duration.Infinite => None
```



Code Correctness: Class Does Not Implement equals	Low
--	------------

Package: akka.cluster

main/scala/akka/cluster/ClusterDaemon.scala, line 415 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

```

416 case d: FiniteDuration =>
417   Some(scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(d), d, self, PublishStatsTick))
418 }
```

test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala, line 32 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: isAvailable()
File: test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala:32
Taint Flags:

```

29
30 override def isAvailable: Boolean = status match {
31   case Unknown | Up => true
32   case Down => false
33 }
34
35 override def isMonitoring: Boolean = status != Unknown
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1469 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: akka\$cluster\$ClusterCoreDaemon\$\$isJoiningToWeaklyUp()
File: main/scala/akka/cluster/ClusterDaemon.scala:1469
Taint Flags:

```

1466 val enoughMembers: Boolean = isMinNrOfMembersFulfilled
1467 def isJoiningToWeaklyUp(m: Member): Boolean =
1468   m.dataCenter == selfDc &&
1469   m.status == Joining &&
1470   enoughMembers &&
1471   membershipState.dcReachabilityExcludingDownedObservers.isReachable(m.uniqueAddress)
1472 val changedMembers = localMembers.collect {
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala, line 31 (Code Correctness: Class Does Not Implement equals)	Low

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: isAvailable()
File: test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala:31
Taint Flags:

```

28 def markNodeAsAvailable(): Unit = status = Up
29
30 override def isAvailable: Boolean = status match {
31   case Unknown | Up => true
32   case Down => false
33 }
34

```

main/scala/akka/cluster/ClusterEvent.scala, line 572 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: applyOrElse()
File: main/scala/akka/cluster/ClusterEvent.scala:572
Taint Flags:

```

569 case m if m.status == Up => MemberUp(m)
570 case m if m.status == Leaving => MemberLeft(m)
571 case m if m.status == Exiting => MemberExited(m)
572 case m if m.status == Down => MemberDowned(m)
573 case m if m.status == PreparingForShutdown => MemberPreparingForShutdown(m)
574 case m if m.status == ReadyForShutdown => MemberReadyForShutdown(m)
575 // no events for other transitions

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1158 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 1158 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** receiveGossip()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1158**Taint Flags:**

1155 publishMembershipState()

1156

1157 val selfStatus = latestGossip.member(selfUniqueAddress).status

1158 if (selfStatus == Exiting && !exitingTasksInProgress) {

1159 // ExitingCompleted will be received via CoordinatedShutdown to continue

1160 // the leaving process. Meanwhile the gossip state is not marked as seen.

1161 exitingTasksInProgress = true

test/scala/akka/cluster/ClusterTestKit.scala, line 150 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** isMemberUp()**File:** test/scala/akka/cluster/ClusterTestKit.scala:150**Taint Flags:**

147 /**

148 * Returns true if the cluster instance for the provided [[ActorSystem]] is [[MemberStatus.Up]].

149 */

150 def isMemberUp(system: ActorSystem): Boolean = Cluster(system).selfMember.status == MemberStatus.Up

151

152 /**

153 * Returns true if the cluster instance for the provided [[ActorSystem]] has be shutdown.

main/scala/akka/cluster/JoinConfigCompatCheckCluster.scala, line 38 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** check()

Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/JoinConfigCompatCheckCluster.scala, line 38 (Code Correctness: Class Does Not Implement equals)	Low

File: main/scala/akka/cluster/JoinConfigCompatCheckCluster.scala:38

Taint Flags:

```

35 val toCheckDowningProvider = toCheck.getString(DowningProviderPath)
36 val actualDowningProvider = actualConfig.getString(DowningProviderPath)
37 val downingProviderResult =
38 if (toCheckDowningProvider == actualDowningProvider || Set(toCheckDowningProvider, actualDowningProvider) == Set(
39 AkkaSbrProviderClass,
40 LightbendSbrProviderClass))
41 Valid

```

main/scala/akka/cluster/ClusterEvent.scala, line 568 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: applyOrElse()

File: main/scala/akka/cluster/ClusterEvent.scala:568

Taint Flags:

```

565 }
566 val memberEvents = (newMembers ++ changedMembers).unsorted.collect {
567 case m if m.status == Joining => MemberJoined(m)
568 case m if m.status == WeaklyUp => MemberWeaklyUp(m)
569 case m if m.status == Up => MemberUp(m)
570 case m if m.status == Leaving => MemberLeft(m)
571 case m if m.status == Exiting => MemberExited(m)

```

main/scala/akka/cluster/ClusterDaemon.scala, line 426 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: preStart()

File: main/scala/akka/cluster/ClusterDaemon.scala:426

Taint Flags:

```

423 cluster.downingProvider.downingActorProps match {

```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 426 (Code Correctness: Class Does Not Implement equals)	Low

```

424 case Some(props) =>
425 val propsWithDispatcher =
426 if (props.dispatcher == Deploy.NoDispatcherGiven) props.withDispatcher(context.props.dispatcher)
427 else props
428
429 context.actorOf(propsWithDispatcher, name = "downingProvider")

```

main/scala/akka/cluster/ClusterEvent.scala, line 574 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: applyOrElse()
File: main/scala/akka/cluster/ClusterEvent.scala:574
Taint Flags:

```

571 case m if m.status == Exiting => MemberExited(m)
572 case m if m.status == Down => MemberDowned(m)
573 case m if m.status == PreparingForShutdown => MemberPreparingForShutdown(m)
574 case m if m.status == ReadyForShutdown => MemberReadyForShutdown(m)
575 // no events for other transitions
576 }
577

```

main/scala/akka/cluster/Reachability.scala, line 146 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: change()
File: main/scala/akka/cluster/Reachability.scala:146
Taint Flags:

```

143 } else
144 new Reachability(records :+ newRecord, newVersions)
145 case Some(oldRecord) =>
146 if (oldRecord.status == Terminated || oldRecord.status == status)

```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/Reachability.scala, line 146 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 147 this 148 else { 149 if (status == Reachable && oldObserverRows.forall { </pre>	
main/scala/akka/cluster/Cluster.scala, line 138 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: checkAutoDownUsage() File: main/scala/akka/cluster/Cluster.scala:138 Taint Flags:	
<pre> 135 } 136 137 private def checkAutoDownUsage(): Unit = { 138 if (settings.DowningProviderClassName == "akka.cluster.AutoDowning" 139 settings.config.hasPath("auto-down-unreachable-after") && settings.config.getString(140 "auto-down-unreachable-after") != "off")) 141 logWarning(</pre>	
main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 332 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: atLeastInUpState() File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:332 Taint Flags:	
<pre> 329 330 // Since we need ordering of oldests guaranteed, we must only look at Up (or Leaving, Exiting...) nodes 331 def atLeastInUpState(m: Member): Boolean = 332 m.status != MemberStatus.WeaklyUp && m.status != MemberStatus.Joining 333 334 def init(335 selfDataCenter: DataCenter, </pre>	

Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 1614 (Code Correctness: Class Does Not Implement equals)	Low

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: publishMembershipState()
File: main/scala/akka/cluster/ClusterDaemon.scala:1614
Taint Flags:

```

1611 gossipLogger.logDebug("New gossip published [{}]", membershipState.latestGossip)
1612
1613 publisher ! PublishChanges(membershipState)
1614 if (PublishStatsInterval == Duration.Zero) publishInternalStats()
1615 }
1616
1617 def publishInternalStats(): Unit = {

```

main/scala/akka/cluster/ClusterEvent.scala, line 336 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: ClusterEvent\$MemberExited()
File: main/scala/akka/cluster/ClusterEvent.scala:336
Taint Flags:

```

333 * when all members have seen the `Exiting` status.
334 */
335 final case class MemberExited(member: Member) extends MemberEvent {
336   if (member.status != Exiting) throw new IllegalArgumentException("Expected Exiting status, got: " + member)
337 }
338
339 /**

```

main/scala/akka/cluster/Cluster.scala, line 140 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/Cluster.scala, line 140 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** checkAutoDownUsage()**File:** main/scala/akka/cluster/Cluster.scala:140**Taint Flags:**

```
137 private def checkAutoDownUsage(): Unit = {  
138   if (settings.DowningProviderClassName == "akka.cluster.AutoDowning" ||  
139     (settings.config.hasPath("auto-down-unreachable-after") && settings.config.getString(  
140       "auto-down-unreachable-after") != "off"))  
141     logWarning(  
142       "auto-down has been removed in Akka 2.6.0. See " +  
143       "https://doc.akka.io/docs/akka/2.6/typed/cluster.html#downing for alternatives.")
```

main/scala/akka/cluster/ClusterDaemon.scala, line 748 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** joining()**File:** main/scala/akka/cluster/ClusterDaemon.scala:748**Taint Flags:**

```
745 "Member with wrong protocol tried to join, but was ignored, expected [{}] but was [{ }]",  
746 selfAddress.protocol,  
747 joiningNode.address.protocol)  
748 else if (joiningNode.address.system != selfAddress.system)  
749   logWarning(  
750     "Member with wrong ActorSystem name tried to join, but was ignored, expected [{}] but was [{ }]",  
751     selfAddress.system,
```

main/scala/akka/cluster/ClusterDaemon.scala, line 777 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** joining()

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 777 (Code Correctness: Class Does Not Implement equals)****Low****File:** main/scala/akka/cluster/ClusterDaemon.scala:777**Taint Flags:**

```
774 "New incarnation of existing member [{ }] is trying to join. " +  
775 "Existing will be removed from the cluster and then new member will be allowed to join.",  
776 m)  
777 if (m.status != Down) {  
778 // we can confirm it as terminated/unreachable immediately  
779 val newReachability = latestGossip.overview.reachability.terminated(selfUniqueAddress, m.uniqueAddress)  
780 val newOverview = latestGossip.overview.copy(reachability = newReachability)
```

main/scala/akka/cluster/MembershipState.scala, line 313 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()**File:** main/scala/akka/cluster/MembershipState.scala:313**Taint Flags:**

```
310 // Fall back to localGossip  
311 if (firstSelection.isEmpty) {  
312 latestGossip.members.toVector.collect {  
313 case m if m.dataCenter == state.selfDc && state.validNodeForGossip(m.uniqueAddress) => m.uniqueAddress  
314 }  
315 } else firstSelection  
316
```

main/scala/akka/cluster/Member.scala, line 45 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** toString()**File:** main/scala/akka/cluster/Member.scala:45**Taint Flags:**

```
42 case _ => false
```



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/Member.scala, line 45 (Code Correctness: Class Does Not Implement equals)****Low**

```
43 }
44 override def toString: String = {
45   s"Member($address, $status${if (dataCenter == ClusterSettings.DefaultDataCenter) "" else s", $dataCenter"}${if (appVersion ==
Version.Zero) ""
46   else s", $appVersion"})"
47 }
48
```

main/scala/akka/cluster/MembershipState.scala, line 111 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** applyOrElse()**File:** main/scala/akka/cluster/MembershipState.scala:111**Taint Flags:**

```
108 * @return reachability for data center nodes, with observations from outside the data center or from downed nodes filtered out
109 */
110 lazy val dcReachabilityExcludingDownedObservers: Reachability = {
111   val membersToExclude = members.collect { case m if m.status == Down || m.dataCenter != selfDc => m.uniqueAddress }
112   overview.reachability
113   .removeObservers(membersToExclude)
114   .remove(members.collect { case m if m.dataCenter != selfDc => m.uniqueAddress })

```

main/scala/akka/cluster/Cluster.scala, line 597 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** logAtLevel()**File:** main/scala/akka/cluster/Cluster.scala:597**Taint Flags:**

```
594 logError(cause, log.format(template, arg1, arg2, arg3))
595
596 private def logAtLevel(logLevel: LogLevel, message: String): Unit = {
597   if (settings.SelfDataCenter == ClusterSettings.DefaultDataCenter)

```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/Cluster.scala, line 597 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 598 log.log(logLevel, "Cluster Node [{}] - {}", selfAddress, message) 599 else 600 log.log(logLevel, "Cluster Node [{}] dc [{}] - {}", selfAddress, settings.SelfDataCenter, message) </pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 1367 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: applyOrElse() File: main/scala/akka/cluster/ClusterDaemon.scala:1367 Taint Flags:	
<pre> 1364 } 1365 m.copyUp(upNumber) 1366 1367 case m if m.dataCenter == selfDc && m.status == Leaving => 1368 // Move LEAVING => EXITING (once we have a convergence on LEAVING) 1369 m.copy(status = Exiting) 1370 </pre>	
main/scala/akka/cluster/Cluster.scala, line 332 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: fillLocal() File: main/scala/akka/cluster/Cluster.scala:332 Taint Flags:	
<pre> 329 330 private def fillLocal(address: Address): Address = { 331 // local address might be used if grabbed from actorRef.path.address 332 if (address.hasLocalScope && address.system == selfAddress.system) selfAddress 333 else address 334 } 335 </pre>	



Code Correctness: Class Does Not Implement equals	Low
--	------------

Package: akka.cluster

main/scala/akka/cluster/Reachability.scala, line 241 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: isReachable()
File: main/scala/akka/cluster/Reachability.scala:241
Taint Flags:

```

238 * @return true if the given observer node can reach the subject node.
239 */
240 def isReachable(observer: UniqueAddress, subject: UniqueAddress): Boolean =
241   status(observer, subject) == Reachable
242
243 /**
244 * @return true if there's no negative (Unreachable, Terminated) observation record at all for

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1468 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: akka\$cluster\$ClusterCoreDaemon\$\$isJoiningToWeaklyUp()
File: main/scala/akka/cluster/ClusterDaemon.scala:1468
Taint Flags:

```

1465
1466 val enoughMembers: Boolean = isMinNrOfMembersFulfilled
1467 def isJoiningToWeaklyUp(m: Member): Boolean =
1468   m.dataCenter == selfDc &&
1469   m.status == Joining &&
1470   enoughMembers &&
1471   membershipState.dcReachabilityExcludingDownedObservers.isReachable(m.uniqueAddress)

```

main/scala/akka/cluster/Reachability.scala, line 130 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/Reachability.scala, line 130 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** change()**File:** main/scala/akka/cluster/Reachability.scala:130**Taint Flags:**

```
127 val newRecord = Record(observer, subject, status, v)
128 observerRows(observer) match {
129 // don't record Reachable observation if nothing has been noted so far
130 case None if status == Reachable => this
131 // otherwise, create new instance including this first observation
132 case None =>
133 new Reachability(records :+ newRecord, newVersions)
```

main/scala/akka/cluster/Reachability.scala, line 139 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** change()**File:** main/scala/akka/cluster/Reachability.scala:139**Taint Flags:**

```
136 case Some(oldObserverRows) =>
137 oldObserverRows.get(subject) match {
138 case None =>
139 if (status == Reachable && oldObserverRows.forall { case (_, r) => r.status == Reachable }) {
140 // FIXME: how should we have gotten into this state?
141 // all Reachable, prune by removing the records of the observer, and bump the version
142 new Reachability(records.filterNot(_._observer == observer), newVersions)
```

main/scala/akka/cluster/Reachability.scala, line 149 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** change()

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/Reachability.scala, line 149 (Code Correctness: Class Does Not Implement equals)****Low****File:** main/scala/akka/cluster/Reachability.scala:149**Taint Flags:**

```
146 if (oldRecord.status == Terminated || oldRecord.status == status)
147 this
148 else {
149 if (status == Reachable && oldObserverRows.forall {
150 case (_, r) => r.status == Reachable || r.subject == subject
151 }) {
152 // all Reachable, prune by removing the records of the observer, and bump the version
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1168 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** receiveGossip()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1168**Taint Flags:**

```
1165 coordShutdown.run(CoordinatedShutdown.ClusterLeavingReason)
1166 }
1167
1168 if (selfStatus == Down && localGossip.member(selfUniqueAddress).status != Down) {
1169 logWarning("Received gossip where this member has been downed, from [{ }]", from.address)
1170 shutdownSelfWhenDown()
1171 }
```

main/scala/akka/cluster/Member.scala, line 72 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** isOlderThan()**File:** main/scala/akka/cluster/Member.scala:72**Taint Flags:**

```
69 */
```



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/Member.scala, line 72 (Code Correctness: Class Does Not Implement equals)****Low**

```
70 @throws[IllegalArgumentException]("if members from different data centers")
71 def isOlderThan(other: Member): Boolean = {
72   if (dataCenter != other.dataCenter)
73     throw new IllegalArgumentException(
74       "Comparing members of different data centers with isOlderThan is not allowed. " +
75       s"$[this] vs. [$other]")
76 }
```

main/scala/akka/cluster/MembershipState.scala, line 118 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: applyOrElse()
File: main/scala/akka/cluster/MembershipState.scala:118
Taint Flags:

```
115 }
116
117 lazy val dcReachabilityNoOutsideNodes: Reachability =
118   overview.reachability.remove(members.collect { case m if m.dataCenter != selfDc => m.uniqueAddress })
119
120 /**
121  * @return Up to `crossDcConnections` oldest members for each DC
122  */
```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 332 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: atLeastInUpState()
File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:332
Taint Flags:

```
329
330 // Since we need ordering of oldests guaranteed, we must only look at Up (or Leaving, Exiting...) nodes
331 def atLeastInUpState(m: Member): Boolean =
332   m.status != MemberStatus.WeaklyUp && m.status != MemberStatus.Joining
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 332 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 333 334 def init(335 selfDataCenter: DataCenter,</pre>	
main/scala/akka/cluster/ClusterEvent.scala, line 573 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: applyOrElse() File: main/scala/akka/cluster/ClusterEvent.scala:573 Taint Flags:	
<pre> 570 case m if m.status == Leaving => MemberLeft(m) 571 case m if m.status == Exiting => MemberExited(m) 572 case m if m.status == Down => MemberDowned(m) 573 case m if m.status == PreparingForShutdown => MemberPreparingForShutdown(m) 574 case m if m.status == ReadyForShutdown => MemberReadyForShutdown(m) 575 // no events for other transitions 576 }</pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 1506 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: akka\$cluster\$ClusterCoreDaemon\$\$isAvailable() File: main/scala/akka/cluster/ClusterDaemon.scala:1506 Taint Flags:	
<pre> 1503 val localMembers = localGossip.members 1504 1505 def isAvailable(member: Member): Boolean = { 1506 if (member.dataCenter == SelfDataCenter) failureDetector.isAvailable(member.address) 1507 else crossDcFailureDetector.isAvailable(member.address) 1508 } 1509</pre>	



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/Member.scala, line 84 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** copy()**File:** main/scala/akka/cluster/Member.scala:84**Taint Flags:**

```
81
82 def copy(status: MemberStatus): Member = {
83   val oldStatus = this.status
84   if (status == oldStatus) this
85   else {
86     require(allowedTransitions(oldStatus)(status), s"Invalid member status transition [ ${this} -> ${status}]")
87     new Member(uniqueAddress, upNumber, status, roles, appVersion)
```

main/scala/akka/cluster/ClusterDaemon.scala, line 694 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** join()**File:** main/scala/akka/cluster/ClusterDaemon.scala:694**Taint Flags:**

```
691 * which will reply with a `Welcome` message.
692 */
693 def join(address: Address): Unit = {
694   if (address.protocol != selfAddress.protocol)
695     logWarning(
696       "Trying to join member with wrong protocol, but was ignored, expected [{}] but was [{ }]",
697       selfAddress.protocol,
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1347 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 1347 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** akka\$cluster\$ClusterCoreDaemon\$\$isJoiningToUp()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1347**Taint Flags:**

1344

1345 val changedMembers = {

1346 val enoughMembers: Boolean = isMinNrOfMembersFulfilled

1347 def isJoiningToUp(m: Member): Boolean = (m.status == Joining || m.status == WeaklyUp) && enoughMembers

1348

1349 latestGossip.members.collect {

1350 var upNumber = 0

main/scala/akka/cluster/Cluster.scala, line 604 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** logAtLevel()**File:** main/scala/akka/cluster/Cluster.scala:604**Taint Flags:**

601 }

602

603 private def logAtLevel(marker: LogMarker, logLevel: LogLevel, message: String): Unit = {

604 if (settings.SelfDataCenter == ClusterSettings.DefaultDataCenter)

605 log.log(marker, logLevel, log.format("Cluster Node [{}] - {}", selfAddress, message))

606 else

607 log.log(

test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala, line 35 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** isMonitoring()

Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala, line 35 (Code Correctness: Class Does Not Implement equals)	Low

File: test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala:35

Taint Flags:

```

32 case Down => false
33 }
34
35 override def isMonitoring: Boolean = status != Unknown
36
37 override def heartbeat(): Unit = status = Up
38

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1137 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: receiveGossip()

File: main/scala/akka/cluster/ClusterDaemon.scala:1137

Taint Flags:

```

1134
1135 gossipLogger.logDebug("Receiving gossip from [{}]", from)
1136
1137 if (comparison == VectorClock.Concurrent && cluster.settings.Debug.VerboseGossipLogging) {
1138 gossipLogger.logDebug(
1139 """"Couldn't establish a causal relationship between "remote" gossip and "local" gossip - Remote[{}] - Local[{}] - merged them into [{}]"""",
1140 remoteGossip,

```

main/scala/akka/cluster/ClusterEvent.scala, line 567 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: applyOrElse()

File: main/scala/akka/cluster/ClusterEvent.scala:567

Taint Flags:



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterEvent.scala, line 567 (Code Correctness: Class Does Not Implement equals)****Low**

```
564 newMember
565 }
566 val memberEvents = (newMembers ++ changedMembers).unsorted.collect {
567 case m if m.status == Joining => MemberJoined(m)
568 case m if m.status == WeaklyUp => MemberWeaklyUp(m)
569 case m if m.status == Up => MemberUp(m)
570 case m if m.status == Leaving => MemberLeft(m)
```

main/scala/akka/cluster/ClusterEvent.scala, line 304 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: ClusterEvent\$MemberWeaklyUp()
File: main/scala/akka/cluster/ClusterEvent.scala:304
Taint Flags:

```
301 * It will be moved to `Up` when convergence is reached.
302 */
303 final case class MemberWeaklyUp(member: Member) extends MemberEvent {
304 if (member.status != WeaklyUp) throw new IllegalArgumentException("Expected WeaklyUp status, got: " + member)
305 }
306
307 /**
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1285 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: shutdownSelfWhenDown()
File: main/scala/akka/cluster/ClusterDaemon.scala:1285
Taint Flags:

```
1282 }
1283
1284 def shutdownSelfWhenDown(): Unit = {
```



Code Correctness: Class Does Not Implement equals	Low
--	------------

Package: akka.cluster

main/scala/akka/cluster/ClusterDaemon.scala, line 1285 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

```

1285 if (latestGossip.member(selfUniqueAddress).status == Down) {
1286 // When all reachable have seen the state this member will shutdown itself when it has
1287 // status Down. The down commands should spread before we shutdown.
1288 val unreachable = membershipState.dcReachability.allUnreachableOrTerminated

```

main/scala/akka/cluster/Gossip.scala, line 193 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: applyOrElse()
File: main/scala/akka/cluster/Gossip.scala:193
Taint Flags:

```

190 }
191
192 lazy val reachabilityExcludingDownedObservers: Reachability = {
193 val downed = members.collect { case m if m.status == Down => m }
194 overview.reachability.removeObservers(downed.map(_.uniqueAddress))
195 }
196

```

main/scala/akka/cluster/ClusterRemoteWatcher.scala, line 127 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: memberRemoved()
File: main/scala/akka/cluster/ClusterRemoteWatcher.scala:127
Taint Flags:

```

124 if (m.address != selfAddress) {
125 clusterNodes -= m.address
126
127 if (previousStatus == MemberStatus.Down) {
128 quarantine(
129 m.address,

```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterRemoteWatcher.scala, line 127 (Code Correctness: Class Does Not Implement equals)	Low

```
130 Some(m.uniqueAddress.longUid),
```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 152 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: addMember()

File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:152

Taint Flags:

```
149 if (CrossDcHeartbeatingState.atLeastInUpState(m)) {
150 // since we only monitor nodes in Up or later states, due to the n-th oldest requirement
151 dataCentersState = dataCentersState.addMember(m)
152 if (verboseHeartbeat && m.dataCenter != selfDataCenter)
153 clusterLogger.logDebug("Register member {} for cross DC heartbeat (will only heartbeat if oldest)", m)
154
155 becomeActiveIfResponsibleForHeartbeat()
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1675 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: akka\$cluster\$OnMemberStatusChangeListener\$\$isTriggered()

File: main/scala/akka/cluster/ClusterDaemon.scala:1675

Taint Flags:

```
1672 }
1673
1674 private def isTriggered(m: Member): Boolean =
1675 m.uniqueAddress == cluster.selfUniqueAddress && m.status == status
1676
1677 }
1678
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster	
main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 343 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: init() File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:343 Taint Flags:	
<pre> 340 // TODO unduplicate this with the logic in MembershipState.ageSortedTopOldestMembersPerDc 341 val groupedByDc = members.filter(atLeastInUpState).groupBy(_.dataCenter) 342 343 if (members.ordering == Member.ageOrdering) { 344 // we already have the right ordering 345 groupedByDc 346 } else { </pre>	
main/scala/akka/cluster/ClusterEvent.scala, line 571 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: applyOrElse() File: main/scala/akka/cluster/ClusterEvent.scala:571 Taint Flags:	
<pre> 568 case m if m.status == WeaklyUp => MemberWeaklyUp(m) 569 case m if m.status == Up => MemberUp(m) 570 case m if m.status == Leaving => MemberLeft(m) 571 case m if m.status == Exiting => MemberExited(m) 572 case m if m.status == Down => MemberDowned(m) 573 case m if m.status == PreparingForShutdown => MemberPreparingForShutdown(m) 574 case m if m.status == ReadyForShutdown => MemberReadyForShutdown(m) </pre>	
test/scala/akka/cluster/FailureDetectorPuppet.scala, line 32 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster**test/scala/akka/cluster/FailureDetectorPuppet.scala, line 32 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** isAvailable()**File:** test/scala/akka/cluster/FailureDetectorPuppet.scala:32**Taint Flags:**

```
29 def markNodeAsAvailable(): Unit = status.set(Up)
```

```
30
```

```
31 override def isAvailable: Boolean = status.get match {
```

```
32 case Unknown | Up => true
```

```
33 case Down => false
```

```
34
```

```
35 }
```

main/scala/akka/cluster/Member.scala, line 201 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** highestPriorityOf()**File:** main/scala/akka/cluster/Member.scala:201**Taint Flags:**

```
198 * Where highest priority is furthest along the membership state machine
```

```
199 */
```

```
200 def highestPriorityOf(m1: Member, m2: Member): Member = {
```

```
201 if (m1.status == m2.status)
```

```
202 // preserve the oldest in case of different upNumber
```

```
203 if (m1.isOlderThan(m2)) m1 else m2
```

```
204 else
```

Package: akka.cluster.protobuf**main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 134 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals

Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.protobuf	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 134 (Code Correctness: Class Does Not Implement equals)	Low

Enclosing Method: fromBinary()

File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:134

Taint Flags:

```

131 case ClusterRouterPoolManifest => deserializeClusterRouterPool(bytes)
132 // needs to stay in 2.6.5 to be able to talk to a 2.5.{3,4} node during rolling upgrade
133 case HeartBeatManifestPre2523 => deserializeHeartBeatAsAddress(bytes)
134 case HeartBeatRspManifest2523 => deserializeHeartBeatRspAsUniqueAddress(bytes)
135 case OldGossipStatusManifest => deserializeGossipStatus(bytes)
136 case OldGossipEnvelopeManifest => deserializeGossipEnvelope(bytes)
137 case OldInitJoinManifest => deserializeInitJoin(bytes)

```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 125 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: fromBinary()

File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:125

Taint Flags:

```

122 case GossipEnvelopeManifest => deserializeGossipEnvelope(bytes)
123 case InitJoinManifest => deserializeInitJoin(bytes)
124 case InitJoinAckManifest => deserializeInitJoinAck(bytes)
125 case InitJoinNackManifest => deserializeInitJoinNack(bytes)
126 case JoinManifest => deserializeJoin(bytes)
127 case WelcomeManifest => deserializeWelcome(bytes)
128 case LeaveManifest => deserializeLeave(bytes)

```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 131 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: fromBinary()

File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:131

Taint Flags:



Code Correctness: Class Does Not Implement equals**Low****Package: akka.cluster.protobuf****main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 131 (Code Correctness: Class Does Not Implement equals)****Low**

```
128 case LeaveManifest => deserializeLeave(bytes)
129 case DownManifest => deserializeDown(bytes)
130 case ExitingConfirmedManifest => deserializeExitingConfirmed(bytes)
131 case ClusterRouterPoolManifest => deserializeClusterRouterPool(bytes)
132 // needs to stay in 2.6.5 to be able to talk to a 2.5.{3,4} node during rolling upgrade
133 case HeartBeatManifestPre2523 => deserializeHeartBeatAsAddress(bytes)
134 case HeartBeatRspManifest2523 => deserializeHeartBeatRspAsUniqueAddress(bytes)
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 123 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: fromBinary()
File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:123
Taint Flags:

```
120 case HeartbeatRspManifest => deserializeHeartBeatResponse(bytes)
121 case GossipStatusManifest => deserializeGossipStatus(bytes)
122 case GossipEnvelopeManifest => deserializeGossipEnvelope(bytes)
123 case InitJoinManifest => deserializeInitJoin(bytes)
124 case InitJoinAckManifest => deserializeInitJoinAck(bytes)
125 case InitJoinNackManifest => deserializeInitJoinNack(bytes)
126 case JoinManifest => deserializeJoin(bytes)
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 142 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: fromBinary()
File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:142
Taint Flags:

```
139 case OldInitJoinNackManifest => deserializeInitJoinNack(bytes)
140 case OldJoinManifest => deserializeJoin(bytes)
141 case OldWelcomeManifest => deserializeWelcome(bytes)
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.protobuf	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 142 (Code Correctness: Class Does Not Implement equals)	Low
142 case OldLeaveManifest => deserializeLeave(bytes) 143 case OldDownManifest => deserializeDown(bytes) 144 case OldExitingConfirmedManifest => deserializeExitingConfirmed(bytes) 145 case OldClusterRouterPoolManifest => deserializeClusterRouterPool(bytes)	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 124 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: fromBinary() File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:124 Taint Flags:	
121 case GossipStatusManifest => deserializeGossipStatus(bytes) 122 case GossipEnvelopeManifest => deserializeGossipEnvelope(bytes) 123 case InitJoinManifest => deserializeInitJoin(bytes) 124 case InitJoinAckManifest => deserializeInitJoinAck(bytes) 125 case InitJoinNackManifest => deserializeInitJoinNack(bytes) 126 case JoinManifest => deserializeJoin(bytes) 127 case WelcomeManifest => deserializeWelcome(bytes)	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 127 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: fromBinary() File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:127 Taint Flags:	
124 case InitJoinAckManifest => deserializeInitJoinAck(bytes) 125 case InitJoinNackManifest => deserializeInitJoinNack(bytes) 126 case JoinManifest => deserializeJoin(bytes) 127 case WelcomeManifest => deserializeWelcome(bytes) 128 case LeaveManifest => deserializeLeave(bytes) 129 case DownManifest => deserializeDown(bytes)	



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.protobuf	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 127 (Code Correctness: Class Does Not Implement equals)	Low

```
130 case ExitingConfirmedManifest => deserializeExitingConfirmed(bytes)
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 261 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: getProtocol()

File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:261

Taint Flags:

```
258 private def getProtocol(address: cm.Address): String = {
259   val p = address.getProtocol
260   val pc = protocolCache
261   if (pc == p) pc
262   else {
263     protocolCache = p
264     p
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 128 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: fromBinary()

File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:128

Taint Flags:

```
125 case InitJoinNackManifest => deserializeInitJoinNack(bytes)
126 case JoinManifest => deserializeJoin(bytes)
127 case WelcomeManifest => deserializeWelcome(bytes)
128 case LeaveManifest => deserializeLeave(bytes)
129 case DownManifest => deserializeDown(bytes)
130 case ExitingConfirmedManifest => deserializeExitingConfirmed(bytes)
131 case ClusterRouterPoolManifest => deserializeClusterRouterPool(bytes)
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.protobuf	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 143 (Code Correctness: Class Does Not Implement equals)	Low

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: fromBinary()
File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:143
Taint Flags:

```

140 case OldJoinManifest => deserializeJoin(bytes)
141 case OldWelcomeManifest => deserializeWelcome(bytes)
142 case OldLeaveManifest => deserializeLeave(bytes)
143 case OldDownManifest => deserializeDown(bytes)
144 case OldExitingConfirmedManifest => deserializeExitingConfirmed(bytes)
145 case OldClusterRouterPoolManifest => deserializeClusterRouterPool(bytes)
146 case _ => throw new IllegalArgumentException(s"Unknown manifest [{manifest}]")

```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 129 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: fromBinary()
File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:129
Taint Flags:

```

126 case JoinManifest => deserializeJoin(bytes)
127 case WelcomeManifest => deserializeWelcome(bytes)
128 case LeaveManifest => deserializeLeave(bytes)
129 case DownManifest => deserializeDown(bytes)
130 case ExitingConfirmedManifest => deserializeExitingConfirmed(bytes)
131 case ClusterRouterPoolManifest => deserializeClusterRouterPool(bytes)
132 // needs to stay in 2.6.5 to be able to talk to a 2.5.{3,4} node during rolling upgrade

```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 122 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.protobuf	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 122 (Code Correctness: Class Does Not Implement equals)	Low
Sink Details	

Sink: FunctionCall: equals
Enclosing Method: fromBinary()
File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:122
Taint Flags:

```

119 case HeartbeatManifest => deserializeHeartBeat(bytes)
120 case HeartbeatRspManifest => deserializeHeartBeatResponse(bytes)
121 case GossipStatusManifest => deserializeGossipStatus(bytes)
122 case GossipEnvelopeManifest => deserializeGossipEnvelope(bytes)
123 case InitJoinManifest => deserializeInitJoin(bytes)
124 case InitJoinAckManifest => deserializeInitJoinAck(bytes)
125 case InitJoinNackManifest => deserializeInitJoinNack(bytes)

```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 126 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details	
Sink: FunctionCall: equals Enclosing Method: fromBinary() File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:126 Taint Flags:	
<pre> 123 case InitJoinManifest => deserializeInitJoin(bytes) 124 case InitJoinAckManifest => deserializeInitJoinAck(bytes) 125 case InitJoinNackManifest => deserializeInitJoinNack(bytes) 126 case JoinManifest => deserializeJoin(bytes) 127 case WelcomeManifest => deserializeWelcome(bytes) 128 case LeaveManifest => deserializeLeave(bytes) 129 case DownManifest => deserializeDown(bytes) </pre>	

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 133 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details	
Sink: FunctionCall: equals Enclosing Method: fromBinary()	



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.protobuf	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 133 (Code Correctness: Class Does Not Implement equals)	Low

File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:133

Taint Flags:

```

130 case ExitingConfirmedManifest => deserializeExitingConfirmed(bytes)
131 case ClusterRouterPoolManifest => deserializeClusterRouterPool(bytes)
132 // needs to stay in 2.6.5 to be able to talk to a 2.5.{3,4} node during rolling upgrade
133 case HeartBeatManifestPre2523 => deserializeHeartBeatAsAddress(bytes)
134 case HeartBeatRspManifest2523 => deserializeHeartBeatRspAsUniqueAddress(bytes)
135 case OldGossipStatusManifest => deserializeGossipStatus(bytes)
136 case OldGossipEnvelopeManifest => deserializeGossipEnvelope(bytes)

```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 408 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: initJoinAckToByteArray()

File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:408

Taint Flags:

```

405 }
406
407 private def initJoinAckToByteArray(address: Address, configCheck: ConfigCheck): Array[Byte] = {
408   if (configCheck == ConfigCheckUnsupportedByJoiningNode)
409     addressToProtoByteArray(address) // plain Address in 2.5.9 or earlier
410   else
411     initJoinAckToProto(address, configCheck).toByteArray

```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 140 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: fromBinary()

File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:140

Taint Flags:

```

137 case OldInitJoinManifest => deserializeInitJoin(bytes)

```



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster.protobuf**main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 140 (Code Correctness: Class Does Not Implement equals)****Low**

```
138 case OldInitJoinAckManifest => deserializeInitJoinAck(bytes)
139 case OldInitJoinNackManifest => deserializeInitJoinNack(bytes)
140 case OldJoinManifest => deserializeJoin(bytes)
141 case OldWelcomeManifest => deserializeWelcome(bytes)
142 case OldLeaveManifest => deserializeLeave(bytes)
143 case OldDownManifest => deserializeDown(bytes)
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 135 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: fromBinary()
File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:135
Taint Flags:

```
132 // needs to stay in 2.6.5 to be able to talk to a 2.5.{3,4} node during rolling upgrade
133 case HeartBeatManifestPre2523 => deserializeHeartBeatAsAddress(bytes)
134 case HeartBeatRspManifest2523 => deserializeHeartBeatRspAsUniqueAddress(bytes)
135 case OldGossipStatusManifest => deserializeGossipStatus(bytes)
136 case OldGossipEnvelopeManifest => deserializeGossipEnvelope(bytes)
137 case OldInitJoinManifest => deserializeInitJoin(bytes)
138 case OldInitJoinAckManifest => deserializeInitJoinAck(bytes)
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 139 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: fromBinary()
File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:139
Taint Flags:

```
136 case OldGossipEnvelopeManifest => deserializeGossipEnvelope(bytes)
137 case OldInitJoinManifest => deserializeInitJoin(bytes)
138 case OldInitJoinAckManifest => deserializeInitJoinAck(bytes)
139 case OldInitJoinNackManifest => deserializeInitJoinNack(bytes)
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.protobuf	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 139 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 140 case OldJoinManifest => deserializeJoin(bytes) 141 case OldWelcomeManifest => deserializeWelcome(bytes) 142 case OldLeaveManifest => deserializeLeave(bytes) </pre>	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 130 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: fromBinary() File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:130 Taint Flags:	
<pre> 127 case WelcomeManifest => deserializeWelcome(bytes) 128 case LeaveManifest => deserializeLeave(bytes) 129 case DownManifest => deserializeDown(bytes) 130 case ExitingConfirmedManifest => deserializeExitingConfirmed(bytes) 131 case ClusterRouterPoolManifest => deserializeClusterRouterPool(bytes) 132 // needs to stay in 2.6.5 to be able to talk to a 2.5.{3,4} node during rolling upgrade 133 case HeartBeatManifestPre2523 => deserializeHeartBeatAsAddress(bytes) </pre>	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 141 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: fromBinary() File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:141 Taint Flags:	
<pre> 138 case OldInitJoinAckManifest => deserializeInitJoinAck(bytes) 139 case OldInitJoinNackManifest => deserializeInitJoinNack(bytes) 140 case OldJoinManifest => deserializeJoin(bytes) 141 case OldWelcomeManifest => deserializeWelcome(bytes) 142 case OldLeaveManifest => deserializeLeave(bytes) 143 case OldDownManifest => deserializeDown(bytes) 144 case OldExitingConfirmedManifest => deserializeExitingConfirmed(bytes) </pre>	



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster.protobuf**main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 144 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** fromBinary()**File:** main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:144**Taint Flags:**

```
141 case OldWelcomeManifest => deserializeWelcome(bytes)
142 case OldLeaveManifest => deserializeLeave(bytes)
143 case OldDownManifest => deserializeDown(bytes)
144 case OldExitingConfirmedManifest => deserializeExitingConfirmed(bytes)
145 case OldClusterRouterPoolManifest => deserializeClusterRouterPool(bytes)
146 case _ => throw new IllegalArgumentException(s"Unknown manifest [{manifest}]")
147 }
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 137 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** fromBinary()**File:** main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:137**Taint Flags:**

```
134 case HeartBeatRspManifest2523 => deserializeHeartBeatRspAsUniqueAddress(bytes)
135 case OldGossipStatusManifest => deserializeGossipStatus(bytes)
136 case OldGossipEnvelopeManifest => deserializeGossipEnvelope(bytes)
137 case OldInitJoinManifest => deserializeInitJoin(bytes)
138 case OldInitJoinAckManifest => deserializeInitJoinAck(bytes)
139 case OldInitJoinNackManifest => deserializeInitJoinNack(bytes)
140 case OldJoinManifest => deserializeJoin(bytes)
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 145 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster.protobuf**main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 145 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** fromBinary()**File:** main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:145**Taint Flags:**

```
142 case OldLeaveManifest => deserializeLeave(bytes)
143 case OldDownManifest => deserializeDown(bytes)
144 case OldExitingConfirmedManifest => deserializeExitingConfirmed(bytes)
145 case OldClusterRouterPoolManifest => deserializeClusterRouterPool(bytes)
146 case _ => throw new IllegalArgumentException(s"Unknown manifest [{manifest}]")
147 }
148
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 138 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** fromBinary()**File:** main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:138**Taint Flags:**

```
135 case OldGossipStatusManifest => deserializeGossipStatus(bytes)
136 case OldGossipEnvelopeManifest => deserializeGossipEnvelope(bytes)
137 case OldInitJoinManifest => deserializeInitJoin(bytes)
138 case OldInitJoinAckManifest => deserializeInitJoinAck(bytes)
139 case OldInitJoinNackManifest => deserializeInitJoinNack(bytes)
140 case OldJoinManifest => deserializeJoin(bytes)
141 case OldWelcomeManifest => deserializeWelcome(bytes)
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 271 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** getSystem()

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster.protobuf**main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 271 (Code Correctness: Class Does Not Implement equals)****Low****File:** main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:271**Taint Flags:**

```
268 private def getSystem(address: cm.Address): String = {  
269   val s = address.getSystem  
270   val sc = systemCache  
271   if (sc == s) sc  
272   else {  
273     systemCache = s  
274     s
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 136 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** fromBinary()**File:** main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:136**Taint Flags:**

```
133 case HeartBeatManifestPre2523 => deserializeHeartBeatAsAddress(bytes)  
134 case HeartBeatRspManifest2523 => deserializeHeartBeatRspAsUniqueAddress(bytes)  
135 case OldGossipStatusManifest => deserializeGossipStatus(bytes)  
136 case OldGossipEnvelopeManifest => deserializeGossipEnvelope(bytes)  
137 case OldInitJoinManifest => deserializeInitJoin(bytes)  
138 case OldInitJoinAckManifest => deserializeInitJoinAck(bytes)  
139 case OldInitJoinNackManifest => deserializeInitJoinNack(bytes)
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 121 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** fromBinary()**File:** main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:121**Taint Flags:**

```
118 def fromBinary(bytes: Array[Byte], manifest: String): AnyRef = manifest match {
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.protobuf	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 121 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 119 case HeartbeatManifest => deserializeHeartBeat(bytes) 120 case HeartbeatRspManifest => deserializeHeartBeatResponse(bytes) 121 case GossipStatusManifest => deserializeGossipStatus(bytes) 122 case GossipEnvelopeManifest => deserializeGossipEnvelope(bytes) 123 case InitJoinManifest => deserializeInitJoin(bytes) 124 case InitJoinAckManifest => deserializeInitJoinAck(bytes) </pre>	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 120 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: fromBinary() File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:120 Taint Flags:	
<pre> 117 118 def fromBinary(bytes: Array[Byte], manifest: String): AnyRef = manifest match { 119 case HeartbeatManifest => deserializeHeartBeat(bytes) 120 case HeartbeatRspManifest => deserializeHeartBeatResponse(bytes) 121 case GossipStatusManifest => deserializeGossipStatus(bytes) 122 case GossipEnvelopeManifest => deserializeGossipEnvelope(bytes) 123 case InitJoinManifest => deserializeInitJoin(bytes) </pre>	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 119 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: fromBinary() File: main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:119 Taint Flags:	
<pre> 116 } 117 118 def fromBinary(bytes: Array[Byte], manifest: String): AnyRef = manifest match { 119 case HeartbeatManifest => deserializeHeartBeat(bytes) </pre>	



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.protobuf	
main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 119 (Code Correctness: Class Does Not Implement equals)	Low

```

120 case HeartbeatRspManifest => deserializeHeartBeatResponse(bytes)
121 case GossipStatusManifest => deserializeGossipStatus(bytes)
122 case GossipEnvelopeManifest => deserializeGossipEnvelope(bytes)

```

Package: akka.cluster.routing	
main/scala/akka/cluster/routing/ClusterRouterConfig.scala, line 105 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: equals()
File: main/scala/akka/cluster/routing/ClusterRouterConfig.scala:105
Taint Flags:

```

102 override def equals(obj: Any): Boolean =
103   obj match {
104     case that: ClusterRouterGroupSettings =>
105       this.totalInstances.equals(that.totalInstances) &&
106       this.routeesPaths.equals(that.routeesPaths) &&
107       this.allowLocalRoutees == that.allowLocalRoutees &&
108       this.useRoles.equals(that.useRoles)

```

main/scala/akka/cluster/routing/ClusterRouterConfig.scala, line 252 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: equals()
File: main/scala/akka/cluster/routing/ClusterRouterConfig.scala:252
Taint Flags:

```

249 override def equals(obj: Any): Boolean =
250   obj match {
251     case that: ClusterRouterPoolSettings =>
252       this.totalInstances.equals(that.totalInstances) &&
253       this.maxInstancesPerNode.equals(that.maxInstancesPerNode) &&
254       this.allowLocalRoutees == that.allowLocalRoutees &&

```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.routing	
main/scala/akka/cluster/routing/ClusterRouterConfig.scala, line 252 (Code Correctness: Class Does Not Implement equals)	Low
255 this.useRoles.equals(that.useRoles)	

main/scala/akka/cluster/routing/ClusterRouterConfig.scala, line 608 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: isAvailable()
File: main/scala/akka/cluster/routing/ClusterRouterConfig.scala:608
Taint Flags:

```

605 }
606
607 def isAvailable(m: Member): Boolean =
608 (m.status == MemberStatus.Up || m.status == MemberStatus.WeaklyUp) &&
609 satisfiesRoles(m.roles) &&
610 (settings.allowLocalRoutees || m.address != cluster.selfAddress)
611

```

main/scala/akka/cluster/routing/ClusterRouterConfig.scala, line 608 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: isAvailable()
File: main/scala/akka/cluster/routing/ClusterRouterConfig.scala:608
Taint Flags:

```

605 }
606
607 def isAvailable(m: Member): Boolean =
608 (m.status == MemberStatus.Up || m.status == MemberStatus.WeaklyUp) &&
609 satisfiesRoles(m.roles) &&
610 (settings.allowLocalRoutees || m.address != cluster.selfAddress)
611

```



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster.routing**main/scala/akka/cluster/routing/ClusterRouterConfig.scala, line 253 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** equals()**File:** main/scala/akka/cluster/routing/ClusterRouterConfig.scala:253**Taint Flags:**

```
250 obj match {  
251   case that: ClusterRouterPoolSettings =>  
252     this.totalInstances.equals(that.totalInstances) &&  
253     this.maxInstancesPerNode.equals(that.maxInstancesPerNode) &&  
254     this.allowLocalRoutees == that.allowLocalRoutees &&  
255     this.useRoles.equals(that.useRoles)  
256   case _ => false
```

Package: akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 32 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** downRemovalMargin()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala:32**Taint Flags:**

```
29 // if down-removal-margin is defined we let it trump stable-after to allow  
30 // for two different values for SBR downing and cluster tool stop/start after downing  
31 val drm = Cluster(system).settings.DownRemovalMargin  
32 if (drm != Duration.Zero) drm  
33 else settings.DowningStableAfter  
34 }  
35
```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 603 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 603 (Code Correctness: Class Does Not Implement equals)****Low****Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** remove()**File:** main/scala/akka/cluster/sbr/SplitBrainResolver.scala:603**Taint Flags:**

```
600 }  
601  
602 def remove(m: Member): Unit = {  
603   if (selfDc == m.dataCenter) {  
604     if (m.uniqueAddress == selfUniqueAddress)  
605       context.stop(self)  
606   } else
```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 488 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** unreachableMember()**File:** main/scala/akka/cluster/sbr/SplitBrainResolver.scala:488**Taint Flags:**

```
485 def isResponsible: Boolean = leader && selfMemberAdded  
486  
487 def unreachableMember(m: Member): Unit = {  
488   if (m.uniqueAddress != selfUniqueAddress && m.dataCenter == selfDc) {  
489     log.debug("SBR unreachableMember [{}]", m)  
490     mutateMemberInfo(resetStable = true) { () =>  
491       strategy.addUnreachable(m)
```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 501 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details**

Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.sbr	
main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 501 (Code Correctness: Class Does Not Implement equals)	Low

Sink: FunctionCall: equals
Enclosing Method: reachableMember()
File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:501
Taint Flags:

```

498 }
499
500 def reachableMember(m: Member): Unit = {
501 if (m.uniqueAddress != selfUniqueAddress && m.dataCenter == selfDc) {
502 log.debug("SBR reachableMember [{ }]", m)
503 mutateMemberInfo(resetStable = true) { () =>
504 strategy.addReachable(m)

```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 332 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: akka\$cluster\$sbr\$SplitBrainResolverBase\$\$tick()
File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:332
Taint Flags:

```

329 decision)
330 actOnDecision(decision)
331 } else {
332 if (decision.acquireDelay == Duration.Zero)
333 acquireLease() // reply message is AcquireLeaseResult
334 else {
335 log.debug("SBR delayed attempt to acquire lease for [{ } ms]", decision.acquireDelay.toMillis)

```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 546 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: addUp()
File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:546
Taint Flags:



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 546 (Code Correctness: Class Does Not Implement equals)****Low**

```
543 }  
544  
545 def addUp(m: Member): Unit = {  
546   if (selfDc == m.dataCenter) {  
547     log.debug("SBR add Up [{ }]", m)  
548     mutateMemberInfo(resetStable = true) { () =>  
549       strategy.add(m)
```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 165 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: addUnreachable()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:165
Taint Flags:

```
162 }  
163  
164 def addUnreachable(m: Member): Unit = {  
165   require(m.dataCenter == selfDc)  
166  
167   add(m)  
168   _unreachable = _unreachable + m.uniqueAddress
```

main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 93 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: leaseMajoritySettings()
File: main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:93
Taint Flags:

```
90  
91 val leaseImplementation = c.getString("lease-implementation")  
92 require(
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.sbr	
main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 93 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 93 leaseImplementation != "", 94 s"akka.cluster.split-brain-resolver.\$LeaseMajorityName.lease-implementation must be defined") 95 96 val acquireLeaseDelayForMinority = </pre>	
main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 570 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: leaving() File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:570 Taint Flags:	
<pre> 567 } 568 569 def leaving(m: Member): Unit = { 570 if (selfDc == m.dataCenter) { 571 log.debug("SBR leaving [{ }]", m) 572 mutateMemberInfo(resetStable = false) { () => 573 strategy.add(m) </pre>	
main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 588 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: addJoining() File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:588 Taint Flags:	
<pre> 585 } 586 587 def addJoining(m: Member): Unit = { 588 if (selfDc == m.dataCenter) { 589 log.debug("SBR add Joining/WeaklyUp [{ }]", m) 590 strategy.add(m) </pre>	



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.sbr	
main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 588 (Code Correctness: Class Does Not Implement equals)	Low
591 }	

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 179 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: add()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:179
Taint Flags:

```
176 }
177
178 def add(m: Member): Unit = {
179   require(m.dataCenter == selfDc)
180
181   removeFromAllMembers(m)
182   _allMembers += m
```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 186 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: remove()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:186
Taint Flags:

```
183 }
184
185 def remove(m: Member): Unit = {
186   require(m.dataCenter == selfDc)
187
188   removeFromAllMembers(m)
189   _unreachable -= m.uniqueAddress
```



Code Correctness: Class Does Not Implement equals	Low
Package: akka.cluster.sbr	
main/scala/akka/cluster/sbr/DowningStrategy.scala, line 172 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: addReachable() File: main/scala/akka/cluster/sbr/DowningStrategy.scala:172 Taint Flags:	
<pre> 169 } 170 171 def addReachable(m: Member): Unit = { 172 require(m.dataCenter == selfDc) 173 174 add(m) 175 _unreachable = _unreachable - m.uniqueAddress </pre>	
main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 263 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: postStop() File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:263 Taint Flags:	
<pre> 260 private var unreachableDataCenters = Set.empty[DataCenter] 261 262 override def postStop(): Unit = { 263 if (releaseLeaseCondition != NoLease) { 264 log.info(265 "SBR is stopped and owns the lease. The lease will not be released until after the " + 266 "lease heartbeat-timeout.") </pre>	
main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 106 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	



Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 106 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** SplitBrainResolver()**File:** main/scala/akka/cluster/sbr/SplitBrainResolver.scala:106**Taint Flags:**

```
103 s"SBR started. Config: strategy [{ }], stable-after [{ }], down-all-when-unstable [{ }], selfUniqueAddress [{ }], selfDc [$selfDc].",
104 Logging.simpleName(strategy.getClass),
105 stableAfter.toCoarsest,
106 if (downAllWhenUnstable == Duration.Zero) "off" else downAllWhenUnstable.toCoarsest,
107 s"${selfUniqueAddress.address}#${selfUniqueAddress.longUid}"
108
109 override def selfUniqueAddress: UniqueAddress = cluster.selfUniqueAddress
```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 579 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** exited()**File:** main/scala/akka/cluster/sbr/SplitBrainResolver.scala:579**Taint Flags:**

```
576 }
577
578 def exited(m: Member): Unit = {
579 if (selfDc == m.dataCenter) {
580 log.debug("SBR exited [{ }]", m)
581 mutateMemberInfo(resetStable = true) { () =>
582 strategy.add(m)
```

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 72 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** tickInterval()

Code Correctness: Class Does Not Implement equals**Low****Package:** akka.cluster.sbr**test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 72 (Code Correctness: Class Does Not Implement equals)****Low****File:** test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:72**Taint Flags:**

```
69
70 // manual ticks used in this test
71 override def tickInterval: FiniteDuration =
72 if (tick == Duration.Zero) super.tickInterval else tick
73
74 // immediate overdue if Duration.Zero is used
75 override def newStableDeadline(): Deadline = super.newStableDeadline() - 1.nanos
```

Package: akka.cluster.testkit**test/scala/akka/cluster/testkit/AutoDown.scala, line 193 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** scheduleUnreachable()**File:** test/scala/akka/cluster/testkit/AutoDown.scala:193**Taint Flags:**

```
190 scheduleUnreachable(m.uniqueAddress)
191
192 def scheduleUnreachable(node: UniqueAddress): Unit = {
193 if (autoDownUnreachableAfter == Duration.Zero) {
194 downOrAddPending(node)
195 } else {
196 val task = scheduler.scheduleOnce(autoDownUnreachableAfter, self, UnreachableTimeout(node))
```

Package: main.scala.akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 1329 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()

Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 1329 (Code Correctness: Class Does Not Implement equals)	Low

File: main/scala/akka/cluster/ClusterDaemon.scala:1329

Taint Flags:

```

1326 val removedUnreachable = for {
1327   node <- membershipState.dcReachability.allUnreachableOrTerminated
1328   m = latestGossip.member(node)
1329   if m.dataCenter == selfDc && removeUnreachableWithMemberStatus(m.status)
1330 } yield m
1331
1332 val removedExitingConfirmed = exitingConfirmed.filter { n =>

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1340 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/ClusterDaemon.scala:1340

Taint Flags:

```

1337 val removedOtherDc =
1338   if (latestGossip.isMultiDc) {
1339     latestGossip.members.filter { m =>
1340       m.dataCenter != selfDc && removeUnreachableWithMemberStatus(m.status)
1341     }
1342   } else
1343     Set.empty[Member]

```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 117 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/ClusterHeartbeat.scala:117

Taint Flags:

```

114 import clusterLogger._

```



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterHeartbeat.scala, line 117 (Code Correctness: Class Does Not Implement equals)	Low

```

115
116 val filterInternalClusterMembers: Member => Boolean =
117   _dataCenter == cluster.selfDataCenter
118
119 var sequenceNr = 0
120

```

main/scala/akka/cluster/MembershipState.scala, line 63 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/MembershipState.scala:63
Taint Flags:

```

60
61 // full convergence needed for first member in a secondary DC
62 val firstMemberInDc =
63   !members.exists(member => member.dataCenter == selfDc && convergenceMemberStatus(member.status))
64
65 // If another member in the data center that is UP or LEAVING and has not seen this gossip or is exiting
66 // convergence cannot be reached. For the first member in a secondary DC all Joining, WeaklyUp, Up or Leaving

```

main/scala/akka/cluster/MembershipState.scala, line 74 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/MembershipState.scala:74
Taint Flags:

```

71 val memberStatus = if (firstMemberInDc) convergenceMemberStatus + Joining + WeaklyUp else convergenceMemberStatus
72 members.exists(
73   member =>
74   (firstMemberInDc || member.dataCenter == selfDc) &&

```



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/MembershipState.scala, line 74 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 75 memberStatus(member.status) && 76 !(latestGossip.seenByNode(member.uniqueAddress) exitingConfirmed(member.uniqueAddress))) 77 }</pre>	
main/scala/akka/cluster/MembershipState.scala, line 159 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: apply() File: main/scala/akka/cluster/MembershipState.scala:159 Taint Flags:	
<pre> 156 } 157 158 def dcMembers: SortedSet[Member] = 159 if (latestGossip.isMultiDc) members.filter(_.dataCenter == selfDc) 160 else members 161 162 def isLeader(node: UniqueAddress): Boolean =</pre>	
main/scala/akka/cluster/MembershipState.scala, line 175 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: apply() File: main/scala/akka/cluster/MembershipState.scala:175 Taint Flags:	
<pre> 172 val reachability = dcReachability 173 174 val reachableMembersInDc = 175 if (reachability.isAllReachable) mbrs.filter(m => m.dataCenter == selfDc && m.status != Down) 176 else 177 mbrs.filter(178 m =></pre>	

Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster**main/scala/akka/cluster/MembershipState.scala, line 179 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/MembershipState.scala:179**Taint Flags:**

```
176 else
177 mbrs.filter(
178 m =>
179 m.dataCenter == selfDc &&
180 m.status != Down &&
181 (reachability.isReachable(m.uniqueAddress) || m.uniqueAddress == selfUniqueAddress))
182 if (reachableMembersInDc.isEmpty) None
```

main/scala/akka/cluster/MembershipState.scala, line 213 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/MembershipState.scala:213**Taint Flags:**

```
210 def gossipTargetsForExitingMembers(exitingMembers: Set[Member]): Set[Member] = {
211 if (exitingMembers.nonEmpty) {
212 val roles = exitingMembers.flatten(_roles).filterNot(_startsWith(ClusterSettings.DcRolePrefix))
213 val membersSortedByAge = latestGossip.members.toList.filter(_dataCenter == selfDc).sorted(Member.ageOrdering)
214 var targets = Set.empty[Member]
215 if (membersSortedByAge.nonEmpty) {
216 targets += membersSortedByAge.head // oldest of all nodes (in DC)
```

main/scala/akka/cluster/ClusterDaemon.scala, line 875 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)

Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 875 (Code Correctness: Class Does Not Implement equals)	Low
Sink Details	

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/ClusterDaemon.scala:875
Taint Flags:

```

872 def leaving(address: Address): Unit = {
873 // only try to update if the node is available (in the member ring)
874 latestGossip.members.find(_address == address).foreach { existingMember =>
875 if (existingMember.status == Joining || existingMember.status == WeaklyUp || existingMember.status == Up || existingMember.status ==
PreparingForShutdown || existingMember.status == ReadyForShutdown) {
876 // mark node as LEAVING
877 val newMembers = latestGossip.members - existingMember + existingMember.copy(status = Leaving)
878 val newGossip = latestGossip.copy(members = newMembers)

```

main/scala/akka/cluster/ClusterEvent.scala, line 234 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details	
Sink: FunctionCall: equals Enclosing Method: apply() File: main/scala/akka/cluster/ClusterEvent.scala:234 Taint Flags:	
<pre> 231 */ 232 @InternalApi 233 private[akka] def isMemberUp(address: Address): Boolean = 234 members.exists(m => m.address == address && m.status == MemberStatus.Up) 235 236 // for binary compatibility (used to be a case class) 237 def copy(</pre>	

main/scala/akka/cluster/ClusterDaemon.scala, line 378 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details	
Sink: FunctionCall: equals	



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 378 (Code Correctness: Class Does Not Implement equals)	Low

Enclosing Method: apply()

File: main/scala/akka/cluster/ClusterDaemon.scala:378

Taint Flags:

```

375 coordShutdown.addTask(CoordinatedShutdown.PhaseClusterExitingDone, "exiting-completed") {
376   val sys = context.system
377   () =>
378   if (Cluster(sys).isTerminated || Cluster(sys).selfMember.status == Down)
379     Future.successful(Done)
380   else {
381     implicit val timeout = Timeout(coordShutdown.timeout(CoordinatedShutdown.PhaseClusterExitingDone))

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1513 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/ClusterDaemon.scala:1513

Taint Flags:

```

1510 val newlyDetectedUnreachableMembers = localMembers.filterNot { member =>
1511   member.uniqueAddress == selfUniqueAddress ||
1512   localOverview.reachability.status(selfUniqueAddress, member.uniqueAddress) == Reachability.Unreachable ||
1513   localOverview.reachability.status(selfUniqueAddress, member.uniqueAddress) == Reachability.Terminated ||
1514   isAvailable(member)
1515 }
1516

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1441 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/ClusterDaemon.scala:1441

Taint Flags:



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 1441 (Code Correctness: Class Does Not Implement equals)	Low

```

1438 if (pruned ne latestGossip) {
1439   updateLatestGossip(pruned)
1440   publishMembershipState()
1441   gossipExitingMembersToOldest(changedMembers.filter(_.status == Exiting))
1442 }
1443 }
1444

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1538 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/ClusterDaemon.scala:1538
Taint Flags:

```

1535
1536 updateLatestGossip(newGossip)
1537
1538 val (exiting, nonExiting) = newlyDetectedUnreachableMembers.partition(_.status == Exiting)
1539 nonExiting.foreach { node =>
1540   logWarning(ClusterLogMarker.unreachable(node.address), "Marking node as UNREACHABLE [{ }].", node)
1541 }

```

main/scala/akka/cluster/CoordinatedShutdownLeave.scala, line 50 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/CoordinatedShutdownLeave.scala:50
Taint Flags:

```

47 done(replyTo)
48 } else if (s.members.exists(m =>
49   m.uniqueAddress == cluster.selfUniqueAddress &&

```



Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster**main/scala/akka/cluster/CoordinatedShutdownLeave.scala, line 50 (Code Correctness: Class Does Not Implement equals)****Low**

```
50 (m.status == Leaving || m.status == Exiting || m.status == Down))) {  
51 done(replyTo)  
52 }  
53 case MemberLeft(m) =>
```

main/scala/akka/cluster/CoordinatedShutdownLeave.scala, line 50 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/CoordinatedShutdownLeave.scala:50**Taint Flags:**

```
47 done(replyTo)  
48 } else if (s.members.exists(m =>  
49 m.uniqueAddress == cluster.selfUniqueAddress &&  
50 (m.status == Leaving || m.status == Exiting || m.status == Down))) {  
51 done(replyTo)  
52 }  
53 case MemberLeft(m) =>
```

main/scala/akka/cluster/ClusterDaemon.scala, line 875 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/ClusterDaemon.scala:875**Taint Flags:**

```
872 def leaving(address: Address): Unit = {  
873 // only try to update if the node is available (in the member ring)  
874 latestGossip.members.find(_.address == address).foreach { existingMember =>  
875 if (existingMember.status == Joining || existingMember.status == WeaklyUp || existingMember.status == Up || existingMember.status ==  
PreparingForShutdown || existingMember.status == ReadyForShutdown) {  
876 // mark node as LEAVING  
877 val newMembers = latestGossip.members - existingMember + existingMember.copy(status = Leaving)
```



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 875 (Code Correctness: Class Does Not Implement equals)	Low

```
878 val newGossip = latestGossip.copy(members = newMembers)
```

main/scala/akka/cluster/ClusterDaemon.scala, line 875 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/ClusterDaemon.scala:875

Taint Flags:

```
872 def leaving(address: Address): Unit = {
873 // only try to update if the node is available (in the member ring)
874 latestGossip.members.find(_.address == address).foreach { existingMember =>
875 if (existingMember.status == Joining || existingMember.status == WeaklyUp || existingMember.status == Up || existingMember.status ==
PreparingForShutdown || existingMember.status == ReadyForShutdown) {
876 // mark node as LEAVING
877 val newMembers = latestGossip.members - existingMember + existingMember.copy(status = Leaving)
878 val newGossip = latestGossip.copy(members = newMembers)
```

main/scala/akka/cluster/Reachability.scala, line 88 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/Reachability.scala:88

Taint Flags:

```
85 mapBuilder += (r.observer -> m)
86
87 if (r.status == Unreachable) allUnreachable += r.subject
88 else if (r.status == Terminated) allTerminated += r.subject
89 }
90
91 val observerRowsMap: Map[UniqueAddress, Map[UniqueAddress, Reachability.Record]] = mapBuilder.toMap
```



Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 1334 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1334**Taint Flags:**

1331

1332 val removedExitingConfirmed = exitingConfirmed.filter { n =>

1333 val member = latestGossip.member(n)

1334 member.dataCenter == selfDc && member.status == Exiting

1335 }

1336

1337 val removedOtherDc =

main/scala/akka/cluster/Reachability.scala, line 139 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/Reachability.scala:139**Taint Flags:**

136 case Some(oldObserverRows) =>

137 oldObserverRows.get(subject) match {

138 case None =>

139 if (status == Reachable && oldObserverRows.forall { case (_, r) => r.status == Reachable }) {

140 // FIXME: how should we have gotten into this state?

141 // all Reachable, prune by removing the records of the observer, and bump the version

142 new Reachability(records.filterNot(_._observer == observer), newVersions)

main/scala/akka/cluster/Reachability.scala, line 150 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)

Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster**main/scala/akka/cluster/Reachability.scala, line 150 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/Reachability.scala:150**Taint Flags:**

```
147 this
148 else {
149 if (status == Reachable && oldObserverRows.forall {
150 case (_, r) => r.status == Reachable || r.subject == subject
151 }) {
152 // all Reachable, prune by removing the records of the observer, and bump the version
153 new Reachability(records.filterNot(_observer == observer), newVersions)
```

main/scala/akka/cluster/MembershipState.scala, line 104 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/MembershipState.scala:104**Taint Flags:**

```
101 */
102 lazy val dcReachabilityWithoutObservationsWithin: Reachability =
103 dcReachability.filterRecords { r =>
104 latestGossip.member(r.subject).dataCenter != selfDc
105 }
106
107 /**
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1334 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()

Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 1334 (Code Correctness: Class Does Not Implement equals)	Low

File: main/scala/akka/cluster/ClusterDaemon.scala:1334

Taint Flags:

```

1331
1332 val removedExitingConfirmed = exitingConfirmed.filter { n =>
1333 val member = latestGossip.member(n)
1334 member.dataCenter == selfDc && member.status == Exiting
1335 }
1336
1337 val removedOtherDc =

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1411 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/ClusterDaemon.scala:1411

Taint Flags:

```

1408 m.status)
1409 }
1410 removedUnreachable.foreach { m =>
1411 val status = if (m.status == Exiting) "exiting" else "unreachable"
1412 logInfo(
1413 ClusterLogMarker.memberChanged(m.uniqueAddress, MemberStatus.Removed),
1414 "Leader is removing {} node [{}]",

```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 58 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:58

Taint Flags:

```

55 var activelyMonitoring = false

```



Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster**main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 58 (Code Correctness: Class Does Not Implement equals)****Low**

```
56
57 val isExternalClusterMember: Member => Boolean =
58 member => member.dataCenter != cluster.selfDataCenter
59
60 val crossDcSettings: cluster.settings.CrossDcFailureDetectorSettings =
61 cluster.settings.MultiDataCenter.CrossDcFailureDetectorSettings
```

main/scala/akka/cluster/Reachability.scala, line 278 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/Reachability.scala:278
Taint Flags:

```
275
276 def observersGroupedByUnreachable: Map[UniqueAddress, Set[UniqueAddress]] = {
277 records.groupBy(_.subject).collect {
278 case (subject, records) if records.exists(_.status == Unreachable) =>
279 val observers: Set[UniqueAddress] =
280 records.iterator.collect { case r if r.status == Unreachable => r.observer }.to(immutable.Set)
281 subject -> observers
```

main/scala/akka/cluster/ClusterDaemon.scala, line 212 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details**

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/ClusterDaemon.scala:212
Taint Flags:

```
209 coordShutdown.addTask(CoordinatedShutdown.PhaseClusterLeave, "leave") {
210 val sys = context.system
211 () =>
212 if (Cluster(sys).isTerminated || Cluster(sys).selfMember.status == Down)
```



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 212 (Code Correctness: Class Does Not Implement equals)	Low

```

213 Future.successful(Done)
214 else {
215 implicit val timeout = Timeout(coordShutdown.timeout(CoordinatedShutdown.PhaseClusterLeave))

```

main/scala/akka/cluster/Gossip.scala, line 81 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/Gossip.scala:81
Taint Flags:

```

78 if (func) throw new IllegalArgumentException(s"$expected, but found [$actual]")
79
80 ifTrueThrow(
81 members.exists(_.status == Removed),
82 expected = s"Live members must not have status [$Removed]",
83 actual = s"${members.filter(_.status == Removed)}")
84

```

main/scala/akka/cluster/Gossip.scala, line 83 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/Gossip.scala:83
Taint Flags:

```

80 ifTrueThrow(
81 members.exists(_.status == Removed),
82 expected = s"Live members must not have status [$Removed]",
83 actual = s"${members.filter(_.status == Removed)}")
84
85 val inReachabilityButNotMember = overview.reachability.allObservers.diff(members.map(_.uniqueAddress))
86 ifTrueThrow(

```



Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 1195 (Code Correctness: Class Does Not Implement equals)****Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1195**Taint Flags:**

1192

1193 def isGossipSpeedupNeeded: Boolean = {

1194 if (latestGossip.isMultiDc) {

1195 latestGossip.members.exists(m => m.status == Down || m.dataCenter == cluster.selfDataCenter) ||

1196 latestGossip.overview.seen

1197 .count(membershipState.isInSameDc) < latestGossip.members.count(_.dataCenter == cluster.selfDataCenter) / 2

1198 } else {

main/scala/akka/cluster/ClusterDaemon.scala, line 1199 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1199**Taint Flags:**

1196 latestGossip.overview.seen

1197 .count(membershipState.isInSameDc) < latestGossip.members.count(_.dataCenter == cluster.selfDataCenter) / 2

1198 } else {

1199 latestGossip.members.exists(m => m.status == Down) ||

1200 latestGossip.overview.seen.size < latestGossip.members.size / 2

1201 }

1202 }

main/scala/akka/cluster/CoordinatedShutdownLeave.scala, line 50 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)

Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster**main/scala/akka/cluster/CoordinatedShutdownLeave.scala, line 50 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/CoordinatedShutdownLeave.scala:50**Taint Flags:**

```
47 done(replyTo)
48 } else if (s.members.exists(m =>
49 m.uniqueAddress == cluster.selfUniqueAddress &&
50 (m.status == Leaving || m.status == Exiting || m.status == Down))) {
51 done(replyTo)
52 }
53 case MemberLeft(m) =>
```

main/scala/akka/cluster/MembershipState.scala, line 175 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/MembershipState.scala:175**Taint Flags:**

```
172 val reachability = dcReachability
173
174 val reachableMembersInDc =
175 if (reachability.isAllReachable) mbrs.filter(m => m.dataCenter == selfDc && m.status != Down)
176 else
177 mbrs.filter(
178 m =>
```

main/scala/akka/cluster/MembershipState.scala, line 180 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()

Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster**main/scala/akka/cluster/MembershipState.scala, line 180 (Code Correctness: Class Does Not Implement equals)****Low****File:** main/scala/akka/cluster/MembershipState.scala:180**Taint Flags:**

```
177 mbrs.filter(  
178 m =>  
179 m.dataCenter == selfDc &&  
180 m.status != Down &&  
181 (reachability.isReachable(m.uniqueAddress) || m.uniqueAddress == selfUniqueAddress))  
182 if (reachableMembersInDc.isEmpty) None  
183 else
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1512 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/ClusterDaemon.scala:1512**Taint Flags:**

```
1509  
1510 val newlyDetectedUnreachableMembers = localMembers.filterNot { member =>  
1511 member.uniqueAddress == selfUniqueAddress ||  
1512 localOverview.reachability.status(selfUniqueAddress, member.uniqueAddress) == Reachability.Unreachable ||  
1513 localOverview.reachability.status(selfUniqueAddress, member.uniqueAddress) == Reachability.Terminated ||  
1514 isAvailable(member)  
1515 }
```

main/scala/akka/cluster/Member.scala, line 135 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/Member.scala:135**Taint Flags:**

```
132 */
```



Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster**main/scala/akka/cluster/Member.scala, line 135 (Code Correctness: Class Does Not Implement equals)****Low**

```
133 private[cluster] val leaderStatusOrdering: Ordering[Member] = Ordering.fromLessThan[Member] { (a, b) =>
134 (a.status, b.status) match {
135 case (as, bs) if as == bs => ordering.compare(a, b) <= 0
136 case (Down, _) => false
137 case (_, Down) => true
138 case (Exiting, _) => false
```

main/scala/akka/cluster/Gossip.scala, line 111 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/Gossip.scala:111**Taint Flags:**

```
108 if (members.size <= 1) false
109 else {
110 val dc1 = members.head.dataCenter
111 members.exists(_.dataCenter != dc1)
112 }
113
114 /**
```

main/scala/akka/cluster/ClusterDaemon.scala, line 875 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/ClusterDaemon.scala:875**Taint Flags:**

```
872 def leaving(address: Address): Unit = {
873 // only try to update if the node is available (in the member ring)
874 latestGossip.members.find(_.address == address).foreach { existingMember =>
875 if (existingMember.status == Joining || existingMember.status == WeaklyUp || existingMember.status == Up || existingMember.status ==
PreparingForShutdown || existingMember.status == ReadyForShutdown) {
```



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 875 (Code Correctness: Class Does Not Implement equals)	Low

```

876 // mark node as LEAVING
877 val newMembers = latestGossip.members - existingMember + existingMember.copy(status = Leaving)
878 val newGossip = latestGossip.copy(members = newMembers)

```

main/scala/akka/cluster/Reachability.scala, line 87 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/Reachability.scala:87
Taint Flags:

```

84 }
85 mapBuilder += (r.observer -> m)
86
87 if (r.status == Unreachable) allUnreachable += r.subject
88 else if (r.status == Terminated) allTerminated += r.subject
89 }
90

```

main/scala/akka/cluster/ClusterDaemon.scala, line 875 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/ClusterDaemon.scala:875
Taint Flags:

```

872 def leaving(address: Address): Unit = {
873 // only try to update if the node is available (in the member ring)
874 latestGossip.members.find(_.address == address).foreach { existingMember =>
875 if (existingMember.status == Joining || existingMember.status == WeaklyUp || existingMember.status == Up || existingMember.status ==
PreparingForShutdown || existingMember.status == ReadyForShutdown) {
876 // mark node as LEAVING
877 val newMembers = latestGossip.members - existingMember + existingMember.copy(status = Leaving)
878 val newGossip = latestGossip.copy(members = newMembers)

```



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 875 (Code Correctness: Class Does Not Implement equals)	Low

main/scala/akka/cluster/ClusterDaemon.scala, line 1195 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/ClusterDaemon.scala:1195
Taint Flags:

```

1192
1193 def isGossipSpeedupNeeded: Boolean = {
1194   if (latestGossip.isMultiDc) {
1195     latestGossip.members.exists(m => m.status == Down || m.dataCenter == cluster.selfDataCenter) ||
1196     latestGossip.overview.seen
1197     .count(membershipState.isInSameDc) < latestGossip.members.count(_.dataCenter == cluster.selfDataCenter) / 2
1198   } else {

```

main/scala/akka/cluster/ClusterDaemon.scala, line 1197 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/ClusterDaemon.scala:1197
Taint Flags:

```

1194   if (latestGossip.isMultiDc) {
1195     latestGossip.members.exists(m => m.status == Down || m.dataCenter == cluster.selfDataCenter) ||
1196     latestGossip.overview.seen
1197     .count(membershipState.isInSameDc) < latestGossip.members.count(_.dataCenter == cluster.selfDataCenter) / 2
1198   } else {
1199     latestGossip.members.exists(m => m.status == Down) ||
1200     latestGossip.overview.seen.size < latestGossip.members.size / 2

```



Code Correctness: Class Does Not Implement equals	Low
--	------------

Package: main.scala.akka.cluster.sbr

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 207 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:207
Taint Flags:

```

204 _reachability
205
206 private def isInSelfDc(node: UniqueAddress): Boolean = {
207   _allMembers.exists(m => m.uniqueAddress == node && m.dataCenter == selfDc)
208 }
209
210 private[sbr] def setReachability(r: Reachability): Unit = {

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 112 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:112
Taint Flags:

```

109 (includingPossiblyUp && m.status == MemberStatus.WeaklyUp) ||
110 (excludingPossiblyExiting && m.status == MemberStatus.Leaving) ||
111 m.status == MemberStatus.Down ||
112 m.status == MemberStatus.Exiting)
113
114 def membersWithRole: immutable.SortedSet[Member] =
115   membersWithRole(includingPossiblyUp = false, excludingPossiblyExiting = false)

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 259 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)



Code Correctness: Class Does Not Implement equals**Low****Package:** main.scala.akka.cluster.sbr**main/scala/akka/cluster/sbr/DowningStrategy.scala, line 259 (Code Correctness: Class Does Not Implement equals)****Low****Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/sbr/DowningStrategy.scala:259**Taint Flags:**

```
256
257 def nodesToDown(decision: Decision = decide()): Set[UniqueAddress] = {
258   val downable = members
259   .filterNot(m => m.status == MemberStatus.Down || m.status == MemberStatus.Exiting)
260   .union(joining)
261   .map(_._uniqueAddress)
262 }
```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 323 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** main/scala/akka/cluster/sbr/DowningStrategy.scala:323**Taint Flags:**

```
320
321 def isAllUnreachableDownOrExiting: Boolean = {
322   _unreachable.isEmpty ||
323   unreachableMembers.forall(m => m.status == MemberStatus.Down || m.status == MemberStatus.Exiting)
324 }
325
326 def reverseDecision(decision: AcquireLeaseDecision): Decision = {
```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 110 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()

Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster.sbr	
main/scala/akka/cluster/sbr/DowningStrategy.scala, line 110 (Code Correctness: Class Does Not Implement equals)	Low

File: main/scala/akka/cluster/sbr/DowningStrategy.scala:110

Taint Flags:

```

107 m =>
108 (!includingPossiblyUp && m.status == MemberStatus.Joining) ||
109 (!includingPossiblyUp && m.status == MemberStatus.WeaklyUp) ||
110 (excludingPossiblyExiting && m.status == MemberStatus.Leaving) ||
111 m.status == MemberStatus.Down ||
112 m.status == MemberStatus.Exiting)
113

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 214 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/sbr/DowningStrategy.scala:214

Taint Flags:

```

211 // skip records with Reachability.Reachable, and skip records related to other DC
212 _reachability = r.filterRecords(
213 record =>
214 (record.status == Reachability.Unreachable || record.status == Reachability.Terminated) &&
215 isInSelfDc(record.observer) && isInSelfDc(record.subject))
216 }
217

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 83 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals

Enclosing Method: apply()

File: main/scala/akka/cluster/sbr/DowningStrategy.scala:83

Taint Flags:

```

80

```



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster.sbr	
main/scala/akka/cluster/sbr/DowningStrategy.scala, line 83 (Code Correctness: Class Does Not Implement equals)	Low

```

81 // all Joining and WeaklyUp members in self DC
82 def joining: immutable.SortedSet[Member] =
83   _allMembers.filter(m => m.status == MemberStatus.Joining || m.status == MemberStatus.WeaklyUp)
84
85 // all members in self DC, both joining and up.
86 @InternalStableApi

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 108 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:108
Taint Flags:

```

105 def members(includingPossiblyUp: Boolean, excludingPossiblyExiting: Boolean): immutable.SortedSet[Member] =
106   _allMembers.filterNot(
107     m =>
108       (!includingPossiblyUp && m.status == MemberStatus.Joining) ||
109       (!includingPossiblyUp && m.status == MemberStatus.WeaklyUp) ||
110       (excludingPossiblyExiting && m.status == MemberStatus.Leaving) ||
111       m.status == MemberStatus.Down ||

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 111 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:111
Taint Flags:

```

108 (!includingPossiblyUp && m.status == MemberStatus.Joining) ||
109 (!includingPossiblyUp && m.status == MemberStatus.WeaklyUp) ||
110 (excludingPossiblyExiting && m.status == MemberStatus.Leaving) ||
111 m.status == MemberStatus.Down ||

```



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster.sbr	
main/scala/akka/cluster/sbr/DowningStrategy.scala, line 111 (Code Correctness: Class Does Not Implement equals)	Low

```

112 m.status == MemberStatus.Exiting)
113
114 def membersWithRole: immutable.SortedSet[Member] =

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 259 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:259
Taint Flags:

```

256
257 def nodesToDown(decision: Decision = decide()): Set[UniqueAddress] = {
258 val downable = members
259 .filterNot(m => m.status == MemberStatus.Down || m.status == MemberStatus.Exiting)
260 .union(joining)
261 .map(_.uniqueAddress)
262

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 323 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:323
Taint Flags:

```

320
321 def isAllUnreachableDownOrExiting: Boolean = {
322 _unreachable.isEmpty ||
323 unreachableMembers.forall(m => m.status == MemberStatus.Down || m.status == MemberStatus.Exiting)
324 }
325
326 def reverseDecision(decision: AcquireLeaseDecision): Decision = {

```



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster.sbr	
main/scala/akka/cluster/sbr/DowningStrategy.scala, line 83 (Code Correctness: Class Does Not Implement equals)	Low

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:83
Taint Flags:

```

80
81 // all Joining and WeaklyUp members in self DC
82 def joining: immutable.SortedSet[Member] =
83   _allMembers.filter(m => m.status == MemberStatus.Joining || m.status == MemberStatus.WeaklyUp)
84
85 // all members in self DC, both joining and up.
86 @InternalStableApi

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 109 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:109
Taint Flags:

```

106 _allMembers.filterNot(
107   m =>
108     (!includingPossiblyUp && m.status == MemberStatus.Joining) ||
109     (!includingPossiblyUp && m.status == MemberStatus.WeaklyUp) ||
110     (excludingPossiblyExiting && m.status == MemberStatus.Leaving) ||
111     m.status == MemberStatus.Down ||
112     m.status == MemberStatus.Exiting)

```

main/scala/akka/cluster/sbr/DowningStrategy.scala, line 214 (Code Correctness: Class Does Not Implement equals)	Low
--	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)



Code Correctness: Class Does Not Implement equals	Low
Package: main.scala.akka.cluster.sbr	
main/scala/akka/cluster/sbr/DowningStrategy.scala, line 214 (Code Correctness: Class Does Not Implement equals)	Low

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/DowningStrategy.scala:214
Taint Flags:

```

211 // skip records with Reachability.Reachable, and skip records related to other DC
212 _reachability = r.filterRecords(
213   record =>
214     (record.status == Reachability.Unreachable || record.status == Reachability.Terminated) &&
215     isInSelfDc(record.observer) && isInSelfDc(record.subject)
216 }
217

```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 631 (Code Correctness: Class Does Not Implement equals)	Low
---	------------

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals
Enclosing Method: apply()
File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:631
Taint Flags:

```

628 private def releaseLease(): Unit = {
629   implicit val ec: ExecutionContext = internalDispatcher
630   strategy.lease.foreach { l =>
631     if (releaseLeaseCondition != NoLease) {
632       log.debug("SBR releasing lease")
633       l.release().recover { case _ => false }.map(ReleaseLeaseResult.apply).pipeTo(self)
634     }
635   }
636 }
637

```

Package: test.scala.akka.cluster	
test/scala/akka/cluster/ClusterTestKit.scala, line 123 (Code Correctness: Class Does Not Implement equals)	Low

Issue Details

Kingdom: API Abuse
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: equals



Code Correctness: Class Does Not Implement equals**Low****Package:** test.scala.akka.cluster**test.scala/akka/cluster/ClusterTestKit.scala, line 123 (Code Correctness: Class Does Not Implement equals)****Low****Enclosing Method:** apply()**File:** test.scala/akka/cluster/ClusterTestKit.scala:123**Taint Flags:**

```
120
121 awaitAssert {
122 actorSystems.foreach { sys =>
123 if (sys != actorSystem && Cluster(sys).selfMember.status == MemberStatus.Up) {
124 // check that it's removed from members
125 if (Cluster(sys).state.members.exists(_uniqueAddress == cluster.selfUniqueAddress))
126 throw new AssertionError(
```

test.scala/akka/cluster/ClusterTestKit.scala, line 123 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** test.scala/akka/cluster/ClusterTestKit.scala:123**Taint Flags:**

```
120
121 awaitAssert {
122 actorSystems.foreach { sys =>
123 if (sys != actorSystem && Cluster(sys).selfMember.status == MemberStatus.Up) {
124 // check that it's removed from members
125 if (Cluster(sys).state.members.exists(_uniqueAddress == cluster.selfUniqueAddress))
126 throw new AssertionError(
```

test.scala/akka/cluster/ClusterTestKit.scala, line 136 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** test.scala/akka/cluster/ClusterTestKit.scala:136**Taint Flags:**

Code Correctness: Class Does Not Implement equals**Low****Package:** test.scala.akka.cluster**test/scala/akka/cluster/ClusterTestKit.scala, line 136 (Code Correctness: Class Does Not Implement equals)****Low**

```
133 shutdown(actorSystem, 10.seconds, verifySystemShutdown = true)
134
135 // remove from internal list
136 actorSystems = actorSystems.filterNot(_ == actorSystem)
137
138 val newConfig = ConfigFactory.parseString(s"")
139 akka.remote.classic.netty.tcp.port = $port
```

test/scala/akka/cluster/ResetSystemMessageSeqNrSpec.scala, line 38 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** test/scala/akka/cluster/ResetSystemMessageSeqNrSpec.scala:38**Taint Flags:**

```
35 Cluster(system).state.members.map(_._uniqueAddress) should ===(
36 Set(Cluster(system).selfUniqueAddress, Cluster(sys2).selfUniqueAddress))
37
38 Cluster(system).state.members.forall(_._status == MemberStatus.Up) shouldBe true
39 }
40 }
41
```

test/scala/akka/cluster/ResetSystemMessageSeqNrSpec.scala, line 73 (Code Correctness: Class Does Not Implement equals)**Low****Issue Details****Kingdom:** API Abuse**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: equals**Enclosing Method:** apply()**File:** test/scala/akka/cluster/ResetSystemMessageSeqNrSpec.scala:73**Taint Flags:**

```
70 Cluster(system).state.members.map(_._uniqueAddress) should ===(
71 Set(Cluster(system).selfUniqueAddress, Cluster(sys3).selfUniqueAddress))
72
```



Code Correctness: Class Does Not Implement equals	Low
Package: test.scala.akka.cluster	
test/scala/akka/cluster/ResetSystemMessageSeqNrSpec.scala, line 73 (Code Correctness: Class Does Not Implement equals)	Low
<pre> 73 Cluster(system).state.members.forall(_status == MemberStatus.Up) shouldBe true 74 } 75 } 76 </pre>	
test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 124 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: apply() File: test/scala/akka/cluster/ClusterDomainEventSpec.scala:124 Taint Flags:	
<pre> 121 122 Set(aUp, bUp, dc2AMemberUp, dc2BMemberUp, dc3AMemberUp, dc3BMemberUp).foreach { member => 123 val otherDc = 124 if (member.dataCenter == ClusterSettings.DefaultDataCenter) Seq("dc2") 125 else Seq() 126 127 diffUnreachableDataCenter(</pre>	
test/scala/akka/cluster/CrossDcHeartbeatSenderSpec.scala, line 47 (Code Correctness: Class Does Not Implement equals)	Low
Issue Details	
Kingdom: API Abuse Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: equals Enclosing Method: apply() File: test/scala/akka/cluster/CrossDcHeartbeatSenderSpec.scala:47 Taint Flags:	
<pre> 44 45 val heartbeatProbe = TestProbe() 46 Cluster(system).join(Cluster(system).selfMember.address) 47 awaitAssert(Cluster(system).selfMember.status == MemberStatus.Up) 48 val underTest = system.actorOf(Props(new TestCrossDcHeartbeatSender(heartbeatProbe))) 49 </pre>	

Code Correctness: Class Does Not Implement equals	Low
Package: test.scala.akka.cluster	
test/scala/akka/cluster/CrossDcHeartbeatSenderSpec.scala, line 47 (Code Correctness: Class Does Not Implement equals)	Low
<div>50 underTest ! CurrentClusterState(</div>	

Code Correctness: Constructor Invokes Overridable Function (337 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.



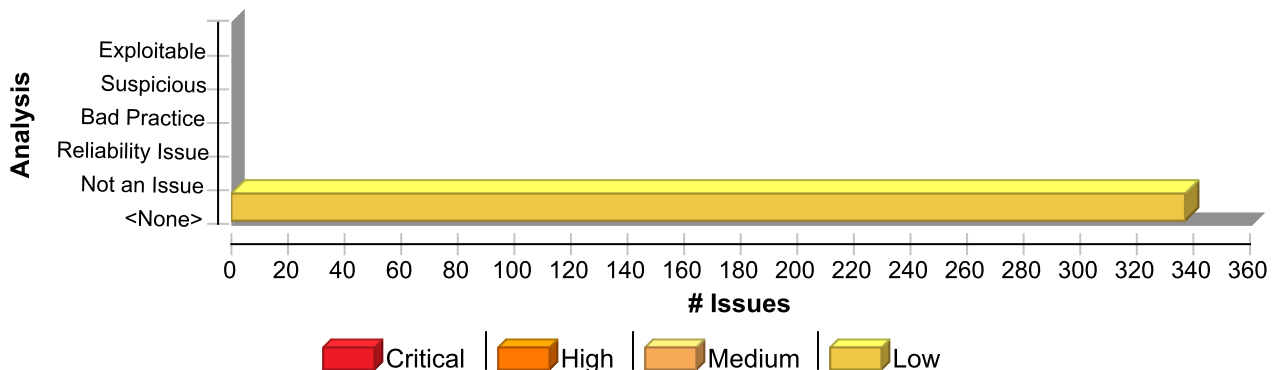
Recommendation

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	337	0	0	337
Total	337	0	0	337

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/VectorClockPerfSpec.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/VectorClockPerfSpec.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function)	Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: lastNode
Enclosing Method: VectorClockPerfSpec()
File: test/scala/akka/cluster/VectorClockPerfSpec.scala:43
Taint Flags:

```

40 val firstNode = nodes.head
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode

```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 45 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: lastNode
Enclosing Method: VectorClockPerfSpec()
File: test/scala/akka/cluster/VectorClockPerfSpec.scala:45
Taint Flags:

```

42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
48 val vcConcurrentMiddle = vcBaseMiddle :+ middleNode

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: config**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:33**Taint Flags:**

```
30 }  
31  
32 class ClusterDomainEventPublisherSpec  
33 extends AkkaSpec(ClusterDomainEventPublisherSpec.config)  
34 with BeforeAndAfterEach  
35 with ImplicitSender {  
36
```

test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: createReachabilityOfSize**Enclosing Method:** ReachabilityPerfSpec()**File:** test/scala/akka/cluster/ReachabilityPerfSpec.scala:42**Taint Flags:**

```
39 }  
40  
41 val reachability1 = createReachabilityOfSize(Reachability.empty, nodesSize)  
42 val reachability2 = createReachabilityOfSize(reachability1, nodesSize)  
43 val reachability3 = addUnreachable(reachability1, nodesSize / 2)  
44 val allowed = reachability1.versions.keySet  
45
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 68 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: g2**Enclosing Method:** ClusterDomainEventPublisherSpec()

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 68 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:68**Taint Flags:**

```
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 69 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: g2**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:69**Taint Flags:**

```
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
```

main/scala/akka/cluster/Gossip.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: emptyMembers**Enclosing Method:** Gossip()**File:** main/scala/akka/cluster/Gossip.scala:23**Taint Flags:**

```
20 private[cluster] object Gossip {
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/Gossip.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
21 type Timestamp = Long
22 val emptyMembers: immutable.SortedSet[Member] = immutable.SortedSet.empty
23 val empty: Gossip = new Gossip(Gossip.emptyMembers)
24
25 def apply(members: immutable.SortedSet[Member]): Gossip =
26 if (members.isEmpty) empty else empty.copy(members = members)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 89 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: eUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:89**Taint Flags:**

```
86 .seen(aUp.uniqueAddress)
87 val state8 = state(g8, aUp.uniqueAddress)
88 val g9 = Gossip(
89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 90 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: eUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:90**Taint Flags:**

```
87 val state8 = state(g8, aUp.uniqueAddress)
88 val g9 = Gossip(
89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 90 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 93 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: eUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:93**Taint Flags:**

```
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
94 overview = GossipOverview(reachability = Reachability.empty))
95 val state10 = state(g10, aUp.uniqueAddress)
96
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 68 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: state**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:68**Taint Flags:**

```
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 72 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: crossDcFailureDetector**Enclosing Method:** CrossDcHeartbeatSender()**File:** main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:72**Taint Flags:**

```
69 ClusterHeartbeatSender.Heartbeat(selfAddress, sequenceNr, System.nanoTime())
70 }
71
72 var dataCentersState: CrossDcHeartbeatingState = CrossDcHeartbeatingState.init(
73 selfDataCenter,
74 crossDcFailureDetector,
75 crossDcSettings.NrOfMonitoringActors,
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 39 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: clockSize**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:39**Taint Flags:**

```
36 // increase for serious measurements
37 val iterations = sys.props.get("akka.cluster.VectorClockPerfSpec.iterations").getOrElse("1000").toInt
38
39 val (vcBefore, nodes) = createVectorClockOfSize(clockSize)
40 val firstNode = nodes.head
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/VectorClockPerfSpec.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: clockSize**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:42**Taint Flags:**

```
39 val (vcBefore, nodes) = createVectorClockOfSize(clockSize)
40 val firstNode = nodes.head
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 66 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: clockSize**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:66**Taint Flags:**

```
63 vc1 == vc2 should == (false)
64 }
65
66 s"VectorClock comparisons of size $clockSize" must {
67
68 s"do a warm up run $iterations times" in {
69 checkThunkFor(vcBaseLast, vcBaseLast, compareTo(Same), iterations)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 45 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: protocol**Enclosing Method:** ClusterDomainEventPublisherSpec()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 45 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:45

Taint Flags:

```

42
43 final val OtherDataCenter = "dc2"
44
45 val aUp = TestMember(Address(protocol, "sys", "a", 2552), Up)
46 val aLeaving = aUp.copy(status = Leaving)
47 val aExiting = aLeaving.copy(status = Exiting)
48 val aRemoved = aExiting.copy(status = Removed)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 49 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: protocol

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:49

Taint Flags:

```

46 val aLeaving = aUp.copy(status = Leaving)
47 val aExiting = aLeaving.copy(status = Exiting)
48 val aRemoved = aExiting.copy(status = Removed)
49 val bExiting = TestMember(Address(protocol, "sys", "b", 2552), Exiting)
50 val bRemoved = bExiting.copy(status = Removed)
51 val cJoining = TestMember(Address(protocol, "sys", "c", 2552), Joining, Set("GRP"))
52 val cUp = cJoining.copy(status = Up)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.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: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:51

Taint Flags:

```

48 val aRemoved = aExiting.copy(status = Removed)

```



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

Package: akka.cluster

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```

49 val bExiting = TestMember(Address(protocol, "sys", "b", 2552), Exiting)
50 val bRemoved = bExiting.copy(status = Removed)
51 val cJoining = TestMember(Address(protocol, "sys", "c", 2552), Joining, Set("GRP"))
52 val cUp = cJoining.copy(status = Up)
53 val cRemoved = cUp.copy(status = Removed)
54 val a51Up = TestMember(Address(protocol, "sys", "a", 2551), Up)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 54 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: protocol
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:54
Taint Flags:

```

51 val cJoining = TestMember(Address(protocol, "sys", "c", 2552), Joining, Set("GRP"))
52 val cUp = cJoining.copy(status = Up)
53 val cRemoved = cUp.copy(status = Removed)
54 val a51Up = TestMember(Address(protocol, "sys", "a", 2551), Up)
55 val dUp = TestMember(Address(protocol, "sys", "d", 2552), Up, Set("GRP"))
56 val eUp = TestMember(Address(protocol, "sys", "e", 2552), Up, Set("GRP"), OtherDataCenter)
57

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 55 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: protocol
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:55
Taint Flags:

```

52 val cUp = cJoining.copy(status = Up)
53 val cRemoved = cUp.copy(status = Removed)
54 val a51Up = TestMember(Address(protocol, "sys", "a", 2551), Up)
55 val dUp = TestMember(Address(protocol, "sys", "d", 2552), Up, Set("GRP"))

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 55 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 56 val eUp = TestMember(Address(protocol, "sys", "e", 2552), Up, Set("GRP"), OtherDataCenter) 57 58 private def state(gossip: Gossip, self: UniqueAddress) = </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 56 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: protocol Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:56 Taint Flags:	
<pre> 53 val cRemoved = cUp.copy(status = Removed) 54 val a51Up = TestMember(Address(protocol, "sys", "a", 2551), Up) 55 val dUp = TestMember(Address(protocol, "sys", "d", 2552), Up, Set("GRP")) 56 val eUp = TestMember(Address(protocol, "sys", "e", 2552), Up, Set("GRP"), OtherDataCenter) 57 58 private def state(gossip: Gossip, self: UniqueAddress) = 59 MembershipState(gossip, self, DefaultDataCenter, crossDcConnections = 5) </pre>	
test/scala/akka/cluster/HeartbeatNodeRingSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cc Enclosing Method: HeartbeatNodeRingSpec() File: test/scala/akka/cluster/HeartbeatNodeRingSpec.scala:21 Taint Flags:	
<pre> 18 val ee = UniqueAddress(Address("akka", "sys", "ee", 2552), 5L) 19 val ff = UniqueAddress(Address("akka", "sys", "ff", 2552), 6L) 20 21 val nodes = Set(aa, bb, cc, dd, ee, ff) 22 23 "A HashedNodeRing" must { 24 </pre>	

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

Package: akka.cluster

test/scala/akka/cluster/HeartbeatNodeRingPerfSpec.scala, line 24 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: createHeartbeatNodeRingOfSize
Enclosing Method: HeartbeatNodeRingPerfSpec()
File: test/scala/akka/cluster/HeartbeatNodeRingPerfSpec.scala:24
Taint Flags:

```

21 HeartbeatNodeRing(selfAddress, nodes.toSet, Set.empty, 5)
22 }
23
24 val heartbeatNodeRing = createHeartbeatNodeRingOfSize(nodesSize)
25
26 private def checkThunkForRing(ring: HeartbeatNodeRing, thunk: HeartbeatNodeRing => Unit, times: Int): Unit =
27   for (_ <- 1 to times) thunk(ring)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 78 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: g5
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:78
Taint Flags:

```

75 .seen(bExiting.uniqueAddress)
76 .seen(cUp.uniqueAddress)
77 .seen(a51Up.uniqueAddress)
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)

```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/VectorClockPerfSpec.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: firstNode**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:44**Taint Flags:**

```
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 47 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: firstNode**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:47**Taint Flags:**

```
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
48 val vcConcurrentMiddle = vcBaseMiddle :+ middleNode
49
50 def checkThunkFor(vc1: VectorClock, vc2: VectorClock, thunk: (VectorClock, VectorClock) => Unit, times: Int): Unit = {
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 30 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: dRoles**Enclosing Method:** ClusterDomainEventSpec()

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 30 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:30**Taint Flags:**

```
27 val cUp = TestMember(Address("akka", "sys", "c", 2552), Up, cRoles)
28 val cLeaving = TestMember(Address("akka", "sys", "c", 2552), Leaving, cRoles)
29 val dRoles = Set("DD", "DE")
30 val dLeaving = TestMember(Address("akka", "sys", "d", 2552), Leaving, dRoles)
31 val dExiting = TestMember(Address("akka", "sys", "d", 2552), Exiting, dRoles)
32 val dRemoved = TestMember(Address("akka", "sys", "d", 2552), Removed, dRoles)
33 val eRoles = Set("EE", "DE")
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 31 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: dRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:31**Taint Flags:**

```
28 val cLeaving = TestMember(Address("akka", "sys", "c", 2552), Leaving, cRoles)
29 val dRoles = Set("DD", "DE")
30 val dLeaving = TestMember(Address("akka", "sys", "d", 2552), Leaving, dRoles)
31 val dExiting = TestMember(Address("akka", "sys", "d", 2552), Exiting, dRoles)
32 val dRemoved = TestMember(Address("akka", "sys", "d", 2552), Removed, dRoles)
33 val eRoles = Set("EE", "DE")
34 val eJoining = TestMember(Address("akka", "sys", "e", 2552), Joining, eRoles)
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 32 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: dRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:32**Taint Flags:**

```
29 val dRoles = Set("DD", "DE")
```



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

Package: akka.cluster

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 32 (Code Correctness: Constructor Invokes Overridable Function)**Low**

```
30 val dLeaving = TestMember(Address("akka", "sys", "d", 2552), Leaving, dRoles)
31 val dExiting = TestMember(Address("akka", "sys", "d", 2552), Exiting, dRoles)
32 val dRemoved = TestMember(Address("akka", "sys", "d", 2552), Removed, dRoles)
33 val eRoles = Set("EE", "DE")
34 val eJoining = TestMember(Address("akka", "sys", "e", 2552), Joining, eRoles)
35 val eUp = TestMember(Address("akka", "sys", "e", 2552), Up, eRoles)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 71 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: a51Up**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:71**Taint Flags:**

```
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: a51Up**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:73**Taint Flags:**

```
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
74 .seen(aUp.uniqueAddress)
75 .seen(bExiting.uniqueAddress)
76 .seen(cUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: a51Up**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:73**Taint Flags:**

```
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)
75 .seen(bExiting.uniqueAddress)
76 .seen(cUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 66 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: g1**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:66**Taint Flags:**

```
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 34 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: eRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:34**Taint Flags:**

```
31 val dExiting = TestMember(Address("akka", "sys", "d", 2552), Exiting, dRoles)
32 val dRemoved = TestMember(Address("akka", "sys", "d", 2552), Removed, dRoles)
33 val eRoles = Set("EE", "DE")
34 val eJoining = TestMember(Address("akka", "sys", "e", 2552), Joining, eRoles)
35 val eUp = TestMember(Address("akka", "sys", "e", 2552), Up, eRoles)
36 val eDown = TestMember(Address("akka", "sys", "e", 2552), Down, eRoles)
37 val selfDummyAddress = UniqueAddress(Address("akka", "sys", "selfDummy", 2552), 17L)
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: eRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:35**Taint Flags:**

```
32 val dRemoved = TestMember(Address("akka", "sys", "d", 2552), Removed, dRoles)
33 val eRoles = Set("EE", "DE")
34 val eJoining = TestMember(Address("akka", "sys", "e", 2552), Joining, eRoles)
35 val eUp = TestMember(Address("akka", "sys", "e", 2552), Up, eRoles)
36 val eDown = TestMember(Address("akka", "sys", "e", 2552), Down, eRoles)
37 val selfDummyAddress = UniqueAddress(Address("akka", "sys", "selfDummy", 2552), 17L)
38
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 36 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 36 (Code Correctness: Constructor Invokes Overridable Function)	Low

Sink Details

Sink: FunctionCall: eRoles
Enclosing Method: ClusterDomainEventSpec()
File: test/scala/akka/cluster/ClusterDomainEventSpec.scala:36
Taint Flags:

```

33 val eRoles = Set("EE", "DE")
34 val eJoining = TestMember(Address("akka", "sys", "e", 2552), Joining, eRoles)
35 val eUp = TestMember(Address("akka", "sys", "e", 2552), Up, eRoles)
36 val eDown = TestMember(Address("akka", "sys", "e", 2552), Down, eRoles)
37 val selfDummyAddress = UniqueAddress(Address("akka", "sys", "selfDummy", 2552), 17L)
38
39 private val originalClusterAssert = sys.props.get("akka.cluster.assert").getOrElse("false")

```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 72 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: crossDcSettings
Enclosing Method: CrossDcHeartbeatSender()
File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:72
Taint Flags:

```

69 ClusterHeartbeatSender.Heartbeat(selfAddress, sequenceNr, System.nanoTime())
70 }
71
72 var dataCentersState: CrossDcHeartbeatingState = CrossDcHeartbeatingState.init(
73 selfDataCenter,
74 crossDcFailureDetector,
75 crossDcSettings.NrOfMonitoringActors,

```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: crossDcSettings
Enclosing Method: CrossDcHeartbeatSender()



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:79

Taint Flags:

```

76 immutable.SortedSet.empty)
77
78 // start periodic heartbeat to other nodes in cluster
79 val heartbeatTask = scheduler.scheduleWithFixedDelay(
80 PeriodicTasksInitialDelay max crossDcSettings.HeartbeatInterval,
81 crossDcSettings.HeartbeatInterval,
82 self,
```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: crossDcSettings

Enclosing Method: CrossDcHeartbeatSender()

File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:79

Taint Flags:

```

76 immutable.SortedSet.empty)
77
78 // start periodic heartbeat to other nodes in cluster
79 val heartbeatTask = scheduler.scheduleWithFixedDelay(
80 PeriodicTasksInitialDelay max crossDcSettings.HeartbeatInterval,
81 crossDcSettings.HeartbeatInterval,
82 self,
```

main/scala/akka/cluster/ClusterSettings.scala, line 153 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: SelfDataCenter

Enclosing Method: ClusterSettings()

File: main/scala/akka/cluster/ClusterSettings.scala:153

Taint Flags:

```

150
```



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

Package: akka.cluster

main/scala/akka/cluster/ClusterSettings.scala, line 153 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

```

151 val SelfDataCenter: DataCenter = cc.getString("multi-data-center.self-data-center")
152
153 val Roles: Set[String] = {
154   val configuredRoles = immutableSeq(cc.getStringList("roles")).toSet.requiring(
155     _.forall(!_._startsWith(DcRolePrefix)),
156     s"Roles must not start with '$DcRolePrefix' as that is reserved for the cluster self-data-center setting")

```

test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 41 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: createReachabilityOfSize
Enclosing Method: ReachabilityPerfSpec()
File: test/scala/akka/cluster/ReachabilityPerfSpec.scala:41
Taint Flags:

```

38 }
39 }
40
41 val reachability1 = createReachabilityOfSize(Reachability.empty, nodesSize)
42 val reachability2 = createReachabilityOfSize(reachability1, nodesSize)
43 val reachability3 = addUnreachable(reachability1, nodesSize / 2)
44 val allowed = reachability1.versions.keySet

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 87 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: state
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:87
Taint Flags:

```

84 members = SortedSet(aUp, bExiting, cUp, dUp),
85 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))
86 .seen(aUp.uniqueAddress)
87 val state8 = state(g8, aUp.uniqueAddress)

```



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

Package: akka.cluster

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 87 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```

88 val g9 = Gossip(
89   members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90   overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))

```

main/scala/akka/cluster/Cluster.scala, line 197 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: joinConfigCompatChecker

Enclosing Method: Cluster()

File: main/scala/akka/cluster/Cluster.scala:197

Taint Flags:

```

194 // create supervisor for daemons under path "/system/cluster"
195 private val clusterDaemons: ActorRef = {
196   system.systemActorOf(
197     Props(classOf[ClusterDaemon], joinConfigCompatChecker).withDispatcher(UseDispatcher).withDeploy(Deploy.local),
198     name = "cluster")
199 }
200

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 82 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: state

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:82

Taint Flags:

```

79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(
84   members = SortedSet(aUp, bExiting, cUp, dUp),
85   overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDeathWatchNotificationSpec.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: config
Enclosing Method: ClusterDeathWatchNotificationSpec()
File: test/scala/akka/cluster/ClusterDeathWatchNotificationSpec.scala:46
Taint Flags:

```

43 }
44
45 class ClusterDeathWatchNotificationSpec
46 extends ArteryMultiNodeSpec(ClusterDeathWatchNotificationSpec.config)
47 with ImplicitSender {
48 import ClusterDeathWatchNotificationSpec.Sender
49

```

test/scala/akka/cluster/GossipSpec.scala, line 25 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: c1
Enclosing Method: GossipSpec()
File: test/scala/akka/cluster/GossipSpec.scala:25
Taint Flags:

```

22 val b1 = TestMember(Address("akka", "sys", "b", 2552), Up)
23 val b2 = TestMember(b1.address, Removed)
24 val c1 = TestMember(Address("akka", "sys", "c", 2552), Leaving)
25 val c2 = TestMember(c1.address, Up)
26 val c3 = TestMember(c1.address, Exiting)
27 val d1 = TestMember(Address("akka", "sys", "d", 2552), Leaving)
28 val e1 = TestMember(Address("akka", "sys", "e", 2552), Joining)

```

test/scala/akka/cluster/GossipSpec.scala, line 26 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/GossipSpec.scala, line 26 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: c1**Enclosing Method:** GossipSpec()**File:** test/scala/akka/cluster/GossipSpec.scala:26**Taint Flags:**

```
23 val b2 = TestMember(b1.address, Removed)
24 val c1 = TestMember(Address("akka", "sys", "c", 2552), Leaving)
25 val c2 = TestMember(c1.address, Up)
26 val c3 = TestMember(c1.address, Exiting)
27 val d1 = TestMember(Address("akka", "sys", "d", 2552), Leaving)
28 val e1 = TestMember(Address("akka", "sys", "e", 2552), Joining)
29 val e2 = TestMember(e1.address, Up)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 87 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: g8**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:87**Taint Flags:**

```
84 members = SortedSet(aUp, bExiting, cUp, dUp),
85 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))
86 .seen(aUp.uniqueAddress)
87 val state8 = state(g8, aUp.uniqueAddress)
88 val g9 = Gossip(
89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 82 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: g7**Enclosing Method:** ClusterDomainEventPublisherSpec()

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 82 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:82**Taint Flags:**

```
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(
84   members = SortedSet(aUp, bExiting, cUp, dUp),
85   overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))
```

test/scala/akka/cluster/GossipSpec.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: b1**Enclosing Method:** GossipSpec()**File:** test/scala/akka/cluster/GossipSpec.scala:23**Taint Flags:**

```
20 val a1 = TestMember(Address("akka", "sys", "a", 2552), Up)
21 val a2 = TestMember(a1.address, Joining)
22 val b1 = TestMember(Address("akka", "sys", "b", 2552), Up)
23 val b2 = TestMember(b1.address, Removed)
24 val c1 = TestMember(Address("akka", "sys", "c", 2552), Leaving)
25 val c2 = TestMember(c1.address, Up)
26 val c3 = TestMember(c1.address, Exiting)
```

test/scala/akka/cluster/ClusterDeathWatchNotificationSpec.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: system2**Enclosing Method:** ClusterDeathWatchNotificationSpec()**File:** test/scala/akka/cluster/ClusterDeathWatchNotificationSpec.scala:53**Taint Flags:**

```
50 private def system1: ActorSystem = system
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDeathWatchNotificationSpec.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
51 private val system2 = newRemoteSystem(name = Some(system.name))
52 private val system3 = newRemoteSystem(name = Some(system.name))
53 private val systems = Vector(system1, system2, system3)
54
55 private val messages = (1 to 100).map(_._toString).toVector
56
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 47 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: vcBaseMiddle**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:47**Taint Flags:**

```
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
48 val vcConcurrentMiddle = vcBaseMiddle :+ middleNode
49
50 def checkThunkFor(vc1: VectorClock, vc2: VectorClock, thunk: (VectorClock, VectorClock) => Unit, times: Int): Unit = {
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: vcBaseMiddle**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:48**Taint Flags:**

```
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
48 val vcConcurrentMiddle = vcBaseMiddle :+ middleNode
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/VectorClockPerfSpec.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function)	Low

49

50 def checkThunkFor(vc1: VectorClock, vc2: VectorClock, thunk: (VectorClock, VectorClock) => Unit, times: Int): Unit = {

51 val vcc1 = copyVectorClock(vc1)

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 84 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: dUp

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:84

Taint Flags:

81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)

82 val state7 = state(g7, aUp.uniqueAddress)

83 val g8 = Gossip(

84 members = SortedSet(aUp, bExiting, cUp, dUp),

85 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))

86 .seen(aUp.uniqueAddress)

87 val state8 = state(g8, aUp.uniqueAddress)

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 85 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: dUp

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:85

Taint Flags:

82 val state7 = state(g7, aUp.uniqueAddress)

83 val g8 = Gossip(

84 members = SortedSet(aUp, bExiting, cUp, dUp),

85 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))

86 .seen(aUp.uniqueAddress)

87 val state8 = state(g8, aUp.uniqueAddress)

88 val g9 = Gossip(



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 89 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: dUp
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:89
Taint Flags:

```

86 .seen(aUp.uniqueAddress)
87 val state8 = state(g8, aUp.uniqueAddress)
88 val g9 = Gossip(
89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 93 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: dUp
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:93
Taint Flags:

```

90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
94 overview = GossipOverview(reachability = Reachability.empty))
95 val state10 = state(g10, aUp.uniqueAddress)
96

```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: bRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:23**Taint Flags:**

```
20 val aUp = TestMember(Address("akka", "sys", "a", 2552), Up, aRoles)
21 val aRemoved = TestMember(Address("akka", "sys", "a", 2552), Removed, aRoles)
22 val bRoles = Set("AB", "BB")
23 val bUp = TestMember(Address("akka", "sys", "b", 2552), Up, bRoles)
24 val bDown = TestMember(Address("akka", "sys", "b", 2552), Down, bRoles)
25 val bRemoved = TestMember(Address("akka", "sys", "b", 2552), Removed, bRoles)
26 val cRoles = Set.empty[String]
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 24 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: bRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:24**Taint Flags:**

```
21 val aRemoved = TestMember(Address("akka", "sys", "a", 2552), Removed, aRoles)
22 val bRoles = Set("AB", "BB")
23 val bUp = TestMember(Address("akka", "sys", "b", 2552), Up, bRoles)
24 val bDown = TestMember(Address("akka", "sys", "b", 2552), Down, bRoles)
25 val bRemoved = TestMember(Address("akka", "sys", "b", 2552), Removed, bRoles)
26 val cRoles = Set.empty[String]
27 val cUp = TestMember(Address("akka", "sys", "c", 2552), Up, cRoles)
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 25 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: bRoles**Enclosing Method:** ClusterDomainEventSpec()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 25 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/ClusterDomainEventSpec.scala:25

Taint Flags:

```

22 val bRoles = Set("AB", "BB")
23 val bUp = TestMember(Address("akka", "sys", "b", 2552), Up, bRoles)
24 val bDown = TestMember(Address("akka", "sys", "b", 2552), Down, bRoles)
25 val bRemoved = TestMember(Address("akka", "sys", "b", 2552), Removed, bRoles)
26 val cRoles = Set.empty[String]
27 val cUp = TestMember(Address("akka", "sys", "c", 2552), Up, cRoles)
28 val cLeaving = TestMember(Address("akka", "sys", "c", 2552), Leaving, cRoles)

```

main/scala/akka/cluster/ClusterSettings.scala, line 149 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: WeaklyUpAfter

Enclosing Method: ClusterSettings()

File: main/scala/akka/cluster/ClusterSettings.scala:149

Taint Flags:

```

146 }
147 }
148
149 val AllowWeaklyUpMembers: Boolean = WeaklyUpAfter != Duration.Zero
150
151 val SelfDataCenter: DataCenter = cc.getString("multi-data-center.self-data-center")
152

```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 353 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodeRing

Enclosing Method: HeartbeatNodeRing()

File: main/scala/akka/cluster/ClusterHeartbeat.scala:353

Taint Flags:

```

350 */

```



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

Package: akka.cluster

main/scala/akka/cluster/ClusterHeartbeat.scala, line 353 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```

351 lazy val myReceivers: Set[UniqueAddress] = receivers(selfAddress)
352
353 private val useAllAsReceivers = monitoredByNrOfMembers >= (nodeRing.size - 1)
354
355 /**
356  * The receivers to use from a specified sender.
```

test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: v2
Enclosing Method: JoinConfigCompatCheckerRollingUpdateSpec()
File: test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala:42
Taint Flags:

```

39 }
40 """)
41
42 val v2Config: Config = v2.withFallback(v1Config)
43
44 val v2ConfigIncompatible: Config = v2.withFallback(baseConfig)
45
```

test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: v2
Enclosing Method: JoinConfigCompatCheckerRollingUpdateSpec()
File: test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala:44
Taint Flags:

```

41
42 val v2Config: Config = v2.withFallback(v1Config)
43
44 val v2ConfigIncompatible: Config = v2.withFallback(baseConfig)
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 45 46 } 47 </pre>	
test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala, line 49 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: v1Config Enclosing Method: JoinConfigCompatCheckerRollingUpdateSpec() File: test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala:49 Taint Flags:	
<pre> 46 } 47 48 class JoinConfigCompatCheckerRollingUpdateSpec 49 extends RollingUpgradeClusterSpec(JoinConfigCompatCheckerRollingUpdateSpec.v1Config) { 50 51 import JoinConfigCompatCheckerRollingUpdateSpec._ 52 </pre>	
test/scala/akka/cluster/ClusterDeathWatchNotificationSpec.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: system1 Enclosing Method: ClusterDeathWatchNotificationSpec() File: test/scala/akka/cluster/ClusterDeathWatchNotificationSpec.scala:53 Taint Flags:	
<pre> 50 private def system1: ActorSystem = system 51 private val system2 = newRemoteSystem(name = Some(system.name)) 52 private val system3 = newRemoteSystem(name = Some(system.name)) 53 private val systems = Vector(system1, system2, system3) 54 55 private val messages = (1 to 100).map(_._toString).toVector 56 </pre>	



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/StartupWithOneThreadSpec.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: config**Enclosing Method:** StartupWithOneThreadSpec()**File:** test/scala/akka/cluster/StartupWithOneThreadSpec.scala:44**Taint Flags:**

```
41 })  
42 }  
43  
44 class StartupWithOneThreadSpec(startTime: Long) extends AkkaSpec(StartupWithOneThreadSpec.config) with ImplicitSender {  
45   import StartupWithOneThreadSpec._  
46  
47   def this() = this(System.nanoTime())
```

main/scala/akka/cluster/ClusterSettings.scala, line 153 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: DcRolePrefix**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:153**Taint Flags:**

```
150  
151 val SelfDataCenter: DataCenter = cc.getString("multi-data-center.self-data-center")  
152  
153 val Roles: Set[String] = {  
154   val configuredRoles = immutableSeq(cc.getStringList("roles")).toSet.requiring(  
155     _.forall(!__.startsWith(DcRolePrefix)),  
156     s"Roles must not start with '$DcRolePrefix' as that is reserved for the cluster self-data-center setting")
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 50 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 50 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: bExiting**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:50**Taint Flags:**

```
47 val aExiting = aLeaving.copy(status = Exiting)
48 val aRemoved = aExiting.copy(status = Removed)
49 val bExiting = TestMember(Address(protocol, "sys", "b", 2552), Exiting)
50 val bRemoved = bExiting.copy(status = Removed)
51 val cJoining = TestMember(Address(protocol, "sys", "c", 2552), Joining, Set("GRP"))
52 val cUp = cJoining.copy(status = Up)
53 val cRemoved = cUp.copy(status = Removed)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 67 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: bExiting**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:67**Taint Flags:**

```
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 69 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: bExiting**Enclosing Method:** ClusterDomainEventPublisherSpec()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 69 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:69

Taint Flags:

```

66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 71 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: bExiting

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:71

Taint Flags:

```

68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: bExiting

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:73

Taint Flags:

```

70 val state3 = state(g3, aUp.uniqueAddress)

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress) 72 val state4 = state(g4, aUp.uniqueAddress) 73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)) 74 .seen(aUp.uniqueAddress) 75 .seen(bExiting.uniqueAddress) 76 .seen(cUp.uniqueAddress) </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: bExiting Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:73 Taint Flags:	
<pre> 70 val state3 = state(g3, aUp.uniqueAddress) 71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress) 72 val state4 = state(g4, aUp.uniqueAddress) 73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)) 74 .seen(aUp.uniqueAddress) 75 .seen(bExiting.uniqueAddress) 76 .seen(cUp.uniqueAddress) </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: bExiting Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:79 Taint Flags:	
<pre> 76 .seen(cUp.uniqueAddress) 77 .seen(a51Up.uniqueAddress) 78 val state5 = state(g5, aUp.uniqueAddress) 79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress) </pre>	



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 80 val state6 = state(g6, aUp.uniqueAddress) 81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress) 82 val state7 = state(g7, aUp.uniqueAddress) </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 81 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: bExiting Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:81 Taint Flags:	
<pre> 78 val state5 = state(g5, aUp.uniqueAddress) 79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress) 80 val state6 = state(g6, aUp.uniqueAddress) 81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress) 82 val state7 = state(g7, aUp.uniqueAddress) 83 val g8 = Gossip(84 members = SortedSet(aUp, bExiting, cUp, dUp), </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 84 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: bExiting Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:84 Taint Flags:	
<pre> 81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress) 82 val state7 = state(g7, aUp.uniqueAddress) 83 val g8 = Gossip(84 members = SortedSet(aUp, bExiting, cUp, dUp), 85 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress))) 86 .seen(aUp.uniqueAddress) 87 val state8 = state(g8, aUp.uniqueAddress) </pre>	

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 89 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: bExiting**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:89**Taint Flags:**

```
86 .seen(aUp.uniqueAddress)
87 val state8 = state(g8, aUp.uniqueAddress)
88 val g9 = Gossip(
89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 93 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: bExiting**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:93**Taint Flags:**

```
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
94 overview = GossipOverview(reachability = Reachability.empty))
95 val state10 = state(g10, aUp.uniqueAddress)
96
```

test/scala/akka/cluster/GossipSpec.scala, line 37 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/GossipSpec.scala, line 37 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: dc2d1**Enclosing Method:** GossipSpec()**File:** test/scala/akka/cluster/GossipSpec.scala:37**Taint Flags:**

```
34 val dc1b1 = TestMember(Address("akka", "sys", "b", 2552), Up, Set.empty, dataCenter = "dc1")
35 val dc2c1 = TestMember(Address("akka", "sys", "c", 2552), Up, Set.empty, dataCenter = "dc2")
36 val dc2d1 = TestMember(Address("akka", "sys", "d", 2552), Up, Set.empty, dataCenter = "dc2")
37 val dc2d2 = TestMember(dc2d1.address, status = Down, roles = Set.empty, dataCenter = dc2d1.dataCenter)
38 // restarted with another uid
39 val dc2d3 =
40 TestMember.withUniqueAddress(UniqueAddress(dc2d1.address, longUid = 3L), Up, Set.empty, dataCenter = "dc2")
```

test/scala/akka/cluster/GossipSpec.scala, line 37 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: dc2d1**Enclosing Method:** GossipSpec()**File:** test/scala/akka/cluster/GossipSpec.scala:37**Taint Flags:**

```
34 val dc1b1 = TestMember(Address("akka", "sys", "b", 2552), Up, Set.empty, dataCenter = "dc1")
35 val dc2c1 = TestMember(Address("akka", "sys", "c", 2552), Up, Set.empty, dataCenter = "dc2")
36 val dc2d1 = TestMember(Address("akka", "sys", "d", 2552), Up, Set.empty, dataCenter = "dc2")
37 val dc2d2 = TestMember(dc2d1.address, status = Down, roles = Set.empty, dataCenter = dc2d1.dataCenter)
38 // restarted with another uid
39 val dc2d3 =
40 TestMember.withUniqueAddress(UniqueAddress(dc2d1.address, longUid = 3L), Up, Set.empty, dataCenter = "dc2")
```

test/scala/akka/cluster/GossipSpec.scala, line 39 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: dc2d1**Enclosing Method:** GossipSpec()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/GossipSpec.scala, line 39 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/GossipSpec.scala:39

Taint Flags:

```

36 val dc2d1 = TestMember(Address("akka", "sys", "d", 2552), Up, Set.empty, dataCenter = "dc2")
37 val dc2d2 = TestMember(dc2d1.address, status = Down, roles = Set.empty, dataCenter = dc2d1.dataCenter)
38 // restarted with another uid
39 val dc2d3 =
40 TestMember.withUniqueAddress(UniqueAddress(dc2d1.address, longUid = 3L), Up, Set.empty, dataCenter = "dc2")
41
42 private def state(g: Gossip, selfMember: Member = a1): MembershipState =

```

main/scala/akka/cluster/ClusterDaemon.scala, line 330 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster

Enclosing Method: ClusterCoreDaemon()

File: main/scala/akka/cluster/ClusterDaemon.scala:330

Taint Flags:

```

327 import cluster.ClusterLogger._
328 import cluster.settings._
329
330 val selfDc = cluster.selfDataCenter
331
332 private val gossipLogger: cluster.ClusterLogger =
333 new cluster.ClusterLogger(

```

main/scala/akka/cluster/ClusterDaemon.scala, line 333 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster

Enclosing Method: ClusterCoreDaemon()

File: main/scala/akka/cluster/ClusterDaemon.scala:333

Taint Flags:

```

330 val selfDc = cluster.selfDataCenter

```



Code Correctness: Constructor Invokes Overridable Function		Low
Package: akka.cluster		
main/scala/akka/cluster/ClusterDaemon.scala, line 333 (Code Correctness: Constructor Invokes Overridable Function)		Low
<pre> 331 332 private val gossipLogger: cluster.ClusterLogger = 333 new cluster.ClusterLogger(334 Logging.withMarker(context.system, ActorWithLogClass(this, ClusterLogClass.ClusterGossip))) 335 336 protected def selfUniqueAddress = cluster.selfUniqueAddress </pre>		
main/scala/akka/cluster/ClusterDaemon.scala, line 339 (Code Correctness: Constructor Invokes Overridable Function)		Low
Issue Details		
Kingdom: Code Quality Scan Engine: SCA (Structural)		
Sink Details		
Sink: FunctionCall: cluster Enclosing Method: ClusterCoreDaemon() File: main/scala/akka/cluster/ClusterDaemon.scala:339 Taint Flags:		
<pre> 336 protected def selfUniqueAddress = cluster.selfUniqueAddress 337 338 val vclockNode = VectorClock.Node(Gossip.vclockName(selfUniqueAddress)) 339 val gossipTargetSelector = new GossipTargetSelector(340 ReduceGossipDifferentViewProbability, 341 cluster.settings.MultiDataCenter.CrossDcGossipProbability) 342 </pre>		
main/scala/akka/cluster/ClusterDaemon.scala, line 345 (Code Correctness: Constructor Invokes Overridable Function)		Low
Issue Details		
Kingdom: Code Quality Scan Engine: SCA (Structural)		
Sink Details		
Sink: FunctionCall: cluster Enclosing Method: ClusterCoreDaemon() File: main/scala/akka/cluster/ClusterDaemon.scala:345 Taint Flags:		
<pre> 342 343 // note that self is not initially member, 344 // and the Gossip is not versioned for this 'Node' yet 345 var membershipState = MembershipState(</pre>		



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 345 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 346 Gossip.empty, 347 cluster.selfUniqueAddress, 348 cluster.settings.SelfDataCenter,</pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 355 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cluster Enclosing Method: ClusterCoreDaemon() File: main/scala/akka/cluster/ClusterDaemon.scala:355 Taint Flags:	
<pre> 352 353 def latestGossip: Gossip = membershipState.latestGossip 354 355 val statsEnabled = PublishStatsInterval.isFinite 356 var gossipStats = GossipStats() 357 358 var seedNodes = SeedNodes</pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 358 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cluster Enclosing Method: ClusterCoreDaemon() File: main/scala/akka/cluster/ClusterDaemon.scala:358 Taint Flags:	
<pre> 355 val statsEnabled = PublishStatsInterval.isFinite 356 var gossipStats = GossipStats() 357 358 var seedNodes = SeedNodes 359 var seedNodeProcess: Option[ActorRef] = None 360 var seedNodeProcessCounter = 0 // for unique names 361 var joinSeedNodesDeadline: Option[Deadline] = None</pre>	



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 396 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** ClusterCoreDaemon()**File:** main/scala/akka/cluster/ClusterDaemon.scala:396**Taint Flags:**

```
393 import context.dispatcher
394
395 // start periodic gossip to random nodes in cluster
396 val gossipTask =
397   scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(GossipInterval), GossipInterval, self, GossipTick)
398
399 // start periodic cluster failure detector reaping (moving nodes condemned by the failure detector to unreachable list)
```

main/scala/akka/cluster/ClusterDaemon.scala, line 396 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** ClusterCoreDaemon()**File:** main/scala/akka/cluster/ClusterDaemon.scala:396**Taint Flags:**

```
393 import context.dispatcher
394
395 // start periodic gossip to random nodes in cluster
396 val gossipTask =
397   scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(GossipInterval), GossipInterval, self, GossipTick)
398
399 // start periodic cluster failure detector reaping (moving nodes condemned by the failure detector to unreachable list)
```

main/scala/akka/cluster/ClusterDaemon.scala, line 396 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterDaemon.scala, line 396 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** ClusterCoreDaemon()**File:** main/scala/akka/cluster/ClusterDaemon.scala:396**Taint Flags:**

393 import context.dispatcher

394

395 // start periodic gossip to random nodes in cluster

396 val gossipTask =

397 scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(GossipInterval), GossipInterval, self, GossipTick)

398

399 // start periodic cluster failure detector reaping (moving nodes condemned by the failure detector to unreachable list)

main/scala/akka/cluster/ClusterDaemon.scala, line 396 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** ClusterCoreDaemon()**File:** main/scala/akka/cluster/ClusterDaemon.scala:396**Taint Flags:**

393 import context.dispatcher

394

395 // start periodic gossip to random nodes in cluster

396 val gossipTask =

397 scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(GossipInterval), GossipInterval, self, GossipTick)

398

399 // start periodic cluster failure detector reaping (moving nodes condemned by the failure detector to unreachable list)

main/scala/akka/cluster/ClusterDaemon.scala, line 400 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** ClusterCoreDaemon()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 400 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: main/scala/akka/cluster/ClusterDaemon.scala:400

Taint Flags:

```

397 scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(GossipInterval), GossipInterval, self, GossipTick)
398
399 // start periodic cluster failure detector reaping (moving nodes condemned by the failure detector to unreachable list)
400 val failureDetectorReaperTask = scheduler.scheduleWithFixedDelay(
401   PeriodicTasksInitialDelay.max(UnreachableNodesReaperInterval),
402   UnreachableNodesReaperInterval,
403   self,
```

main/scala/akka/cluster/ClusterDaemon.scala, line 400 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster

Enclosing Method: ClusterCoreDaemon()

File: main/scala/akka/cluster/ClusterDaemon.scala:400

Taint Flags:

```

397 scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(GossipInterval), GossipInterval, self, GossipTick)
398
399 // start periodic cluster failure detector reaping (moving nodes condemned by the failure detector to unreachable list)
400 val failureDetectorReaperTask = scheduler.scheduleWithFixedDelay(
401   PeriodicTasksInitialDelay.max(UnreachableNodesReaperInterval),
402   UnreachableNodesReaperInterval,
403   self,
```

main/scala/akka/cluster/ClusterDaemon.scala, line 407 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster

Enclosing Method: ClusterCoreDaemon()

File: main/scala/akka/cluster/ClusterDaemon.scala:407

Taint Flags:

```

404 ReapUnreachableTick)
```



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

Package: akka.cluster

main/scala/akka/cluster/ClusterDaemon.scala, line 407 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

```

405
406 // start periodic leader action management (only applies for the current leader)
407 val leaderActionsTask = scheduler.scheduleWithFixedDelay(
408   PeriodicTasksInitialDelay.max(LeaderActionsInterval),
409   LeaderActionsInterval,
410   self,
```

main/scala/akka/cluster/ClusterDaemon.scala, line 407 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterCoreDaemon()
File: main/scala/akka/cluster/ClusterDaemon.scala:407
Taint Flags:

```

404 ReapUnreachableTick)
405
406 // start periodic leader action management (only applies for the current leader)
407 val leaderActionsTask = scheduler.scheduleWithFixedDelay(
408   PeriodicTasksInitialDelay.max(LeaderActionsInterval),
409   LeaderActionsInterval,
410   self,
```

main/scala/akka/cluster/ClusterDaemon.scala, line 407 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterCoreDaemon()
File: main/scala/akka/cluster/ClusterDaemon.scala:407
Taint Flags:

```

404 ReapUnreachableTick)
405
406 // start periodic leader action management (only applies for the current leader)
407 val leaderActionsTask = scheduler.scheduleWithFixedDelay(
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 407 (Code Correctness: Constructor Invokes Overridable Function)	Low

```

408 PeriodicTasksInitialDelay.max(LeaderActionsInterval),
409 LeaderActionsInterval,
410 self,

```

main/scala/akka/cluster/ClusterDaemon.scala, line 414 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterCoreDaemon()
File: main/scala/akka/cluster/ClusterDaemon.scala:414
Taint Flags:

```

411 LeaderActionsTick)
412
413 // start periodic publish of current stats
414 val publishStatsTask: Option[Cancellable] = PublishStatsInterval match {
415 case Duration.Zero | _: Duration.Infinite => None
416 case d: FiniteDuration =>
417   Some(scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(d), d, self, PublishStatsTick))

```

main/scala/akka/cluster/ClusterDaemon.scala, line 417 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterCoreDaemon()
File: main/scala/akka/cluster/ClusterDaemon.scala:417
Taint Flags:

```

414 val publishStatsTask: Option[Cancellable] = PublishStatsInterval match {
415 case Duration.Zero | _: Duration.Infinite => None
416 case d: FiniteDuration =>
417   Some(scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(d), d, self, PublishStatsTick))
418 }
419
420 override def preStart(): Unit = {

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterDaemon.scala, line 417 (Code Correctness: Constructor Invokes Overridable Function)	Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterCoreDaemon()
File: main/scala/akka/cluster/ClusterDaemon.scala:417
Taint Flags:

```

414 val publishStatsTask: Option[Cancellable] = PublishStatsInterval match {
415   case Duration.Zero | _: Duration.Infinite => None
416   case d: FiniteDuration =>
417     Some(scheduler.scheduleWithFixedDelay(PeriodicTasksInitialDelay.max(d), d, self, PublishStatsTick))
418 }
419
420 override def preStart(): Unit = {

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 64 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: state
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:64
Taint Flags:

```

61 val emptyMembershipState = state(Gossip.empty, aUp.uniqueAddress)
62
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 61 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 61 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: state**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:61**Taint Flags:**

```
58 private def state(gossip: Gossip, self: UniqueAddress) =  
59 MembershipState(gossip, self, DefaultDataCenter, crossDcConnections = 5)  
60  
61 val emptyMembershipState = state(Gossip.empty, aUp.uniqueAddress)  
62  
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)  
64 val state0 = state(g0, aUp.uniqueAddress)
```

test/scala/akka/cluster/HeartbeatNodeRingSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: ee**Enclosing Method:** HeartbeatNodeRingSpec()**File:** test/scala/akka/cluster/HeartbeatNodeRingSpec.scala:21**Taint Flags:**

```
18 val ee = UniqueAddress(Address("akka", "sys", "ee", 2552), 5L)  
19 val ff = UniqueAddress(Address("akka", "sys", "ff", 2552), 6L)  
20  
21 val nodes = Set(aa, bb, cc, dd, ee, ff)  
22  
23 "A HashedNodeRing" must {  
24
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 80 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: g6**Enclosing Method:** ClusterDomainEventPublisherSpec()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 80 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:80

Taint Flags:

```

77 .seen(a51Up.uniqueAddress)
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(

```

main/scala/akka/cluster/Cluster.scala, line 112 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ClusterLogger

Enclosing Method: Cluster()

File: main/scala/akka/cluster/Cluster.scala:112

Taint Flags:

```

109 // ClusterJmx is initialized as the last thing in the constructor
110 private var clusterJmx: Option[ClusterJmx] = None
111
112 logInfo("Starting up, Akka version [{}] ...", system.settings.ConfigVersion)
113
114 val failureDetector: FailureDetectorRegistry[Address] = {
115 val createFailureDetector = () =>

```

main/scala/akka/cluster/Cluster.scala, line 155 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ClusterLogger

Enclosing Method: Cluster()

File: main/scala/akka/cluster/Cluster.scala:155

Taint Flags:

```

152 */

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/Cluster.scala, line 155 (Code Correctness: Constructor Invokes Overridable Function)	Low

```

153 private[cluster] val scheduler: Scheduler = {
154   if (system.scheduler.maxFrequency < 1.second / SchedulerTickDuration) {
155     logInfo(
156       "Using a dedicated scheduler for cluster. Default scheduler can be used if configured " +
157       "with 'akka.scheduler.tick-duration' [{ } ms] <= 'akka.cluster.scheduler.tick-duration' [{ } ms].",
158       (1000 / system.scheduler.maxFrequency).toInt,

```

main/scala/akka/cluster/Cluster.scala, line 230 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ClusterLogger
Enclosing Method: Cluster()
File: main/scala/akka/cluster/Cluster.scala:230
Taint Flags:

```

227 Some(jmx)
228 }
229
230 logInfo("Started up successfully")
231
232 // =====
233 // ===== PUBLIC API =====

```

test/scala/akka/cluster/HeartbeatNodeRingPerfSpec.scala, line 24 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodesSize
Enclosing Method: HeartbeatNodeRingPerfSpec()
File: test/scala/akka/cluster/HeartbeatNodeRingPerfSpec.scala:24
Taint Flags:

```

21 HeartbeatNodeRing(selfAddress, nodes.toSet, Set.empty, 5)
22 }
23
24 val heartbeatNodeRing = createHeartbeatNodeRingOfSize(nodesSize)

```



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

Package: akka.cluster

test/scala/akka/cluster/HeartbeatNodeRingPerfSpec.scala, line 24 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```

25
26 private def checkThunkForRing(ring: HeartbeatNodeRing, thunk: HeartbeatNodeRing => Unit, times: Int): Unit =
27   for (_ <- 1 to times) thunk(ring)

```

test/scala/akka/cluster/HeartbeatNodeRingPerfSpec.scala, line 34 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodesSize
Enclosing Method: HeartbeatNodeRingPerfSpec()
File: test/scala/akka/cluster/HeartbeatNodeRingPerfSpec.scala:34
Taint Flags:

```

31 r.myReceivers.isEmpty should ===(false)
32 }
33
34 s"HeartbeatNodeRing of size $nodesSize" must {
35
36 s"do a warm up run, $iterations times" in {
37   checkThunkForRing(heartbeatNodeRing, myReceivers, iterations)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: aUp
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:46
Taint Flags:

```

43 final val OtherDataCenter = "dc2"
44
45 val aUp = TestMember(Address(protocol, "sys", "a", 2552), Up)
46 val aLeaving = aUp.copy(status = Leaving)
47 val aExiting = aLeaving.copy(status = Exiting)
48 val aRemoved = aExiting.copy(status = Removed)
49 val bExiting = TestMember(Address(protocol, "sys", "b", 2552), Exiting)

```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 61 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:61**Taint Flags:**

```
58 private def state(gossip: Gossip, self: UniqueAddress) =  
59 MembershipState(gossip, self, DefaultDataCenter, crossDcConnections = 5)  
60  
61 val emptyMembershipState = state(Gossip.empty, aUp.uniqueAddress)  
62  
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)  
64 val state0 = state(g0, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 63 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:63**Taint Flags:**

```
60  
61 val emptyMembershipState = state(Gossip.empty, aUp.uniqueAddress)  
62  
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)  
64 val state0 = state(g0, aUp.uniqueAddress)  
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)  
66 val state1 = state(g1, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 63 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 63 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:63**Taint Flags:**

```
60
61 val emptyMembershipState = state(Gossip.empty, aUp.uniqueAddress)
62
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 64 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:64**Taint Flags:**

```
61 val emptyMembershipState = state(Gossip.empty, aUp.uniqueAddress)
62
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 65 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 65 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:65**Taint Flags:**

```
62
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 65 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:65**Taint Flags:**

```
62
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 66 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:66**Taint Flags:**

```
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)
```



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

Package: akka.cluster

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 66 (Code Correctness: Constructor Invokes Overridable Function)**Low**

```
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 67 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:67**Taint Flags:**

```
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 67 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:67**Taint Flags:**

```
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
```



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

Package: akka.cluster

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 67 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```

68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 68 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: aUp

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:68

Taint Flags:

```

65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 70 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: aUp

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:70

Taint Flags:

```

67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))

```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 71 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:71**Taint Flags:**

```
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 71 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:71**Taint Flags:**

```
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 72 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 72 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:72**Taint Flags:**

```
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)
75 .seen(bExiting.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:73**Taint Flags:**

```
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)
75 .seen(bExiting.uniqueAddress)
76 .seen(cUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:73

Taint Flags:

```

70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)
75 .seen(bExiting.uniqueAddress)
76 .seen(cUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 78 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: aUp

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:78

Taint Flags:

```

75 .seen(bExiting.uniqueAddress)
76 .seen(cUp.uniqueAddress)
77 .seen(a51Up.uniqueAddress)
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: aUp

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:79

Taint Flags:

```

76 .seen(cUp.uniqueAddress)

```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
77 .seen(a51Up.uniqueAddress)
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 80 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:80**Taint Flags:**

```
77 .seen(a51Up.uniqueAddress)
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 81 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:81**Taint Flags:**

```
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 81 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(
84 members = SortedSet(aUp, bExiting, cUp, dUp),
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 82 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:82**Taint Flags:**

```
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(
84 members = SortedSet(aUp, bExiting, cUp, dUp),
85 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 83 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:83**Taint Flags:**

```
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(
84 members = SortedSet(aUp, bExiting, cUp, dUp),
85 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))
86 .seen(aUp.uniqueAddress)
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 84 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:84**Taint Flags:**

```
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(
84   members = SortedSet(aUp, bExiting, cUp, dUp),
85   overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))
86   .seen(aUp.uniqueAddress)
87 val state8 = state(g8, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 85 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:85**Taint Flags:**

```
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(
84   members = SortedSet(aUp, bExiting, cUp, dUp),
85   overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))
86   .seen(aUp.uniqueAddress)
87 val state8 = state(g8, aUp.uniqueAddress)
88 val g9 = Gossip(
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 87 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 87 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:87**Taint Flags:**

```
84 members = SortedSet(aUp, bExiting, cUp, dUp),
85 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress)))
86 .seen(aUp.uniqueAddress)
87 val state8 = state(g8, aUp.uniqueAddress)
88 val g9 = Gossip(
89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 89 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:89**Taint Flags:**

```
86 .seen(aUp.uniqueAddress)
87 val state8 = state(g8, aUp.uniqueAddress)
88 val g9 = Gossip(
89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 90 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 90 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:90**Taint Flags:**

```
87 val state8 = state(g8, aUp.uniqueAddress)
88 val g9 = Gossip(
89   members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90   overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
93   members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 91 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:91**Taint Flags:**

```
88 val g9 = Gossip(
89   members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
90   overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
93   members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
94   overview = GossipOverview(reachability = Reachability.empty))
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 93 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:93**Taint Flags:**

```
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 93 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
91 val state9 = state(g9, aUp.uniqueAddress)
92 val g10 = Gossip(
93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
94 overview = GossipOverview(reachability = Reachability.empty))
95 val state10 = state(g10, aUp.uniqueAddress)
96
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 95 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:95**Taint Flags:**

```
92 val g10 = Gossip(
93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),
94 overview = GossipOverview(reachability = Reachability.empty))
95 val state10 = state(g10, aUp.uniqueAddress)
96
97 // created in beforeEach
98 var memberSubscriber: TestProbe = _
```

test/scala/akka/cluster/FailureDetectorPuppet.scala, line 25 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: Unknown**Enclosing Method:** FailureDetectorPuppet()**File:** test/scala/akka/cluster/FailureDetectorPuppet.scala:25**Taint Flags:**

```
22 object Down extends Status
23 object Unknown extends Status
24
25 private val status: AtomicReference[Status] = new AtomicReference(Unknown)
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/FailureDetectorPuppet.scala, line 25 (Code Correctness: Constructor Invokes Overridable Function)	Low

```

26
27 def markNodeAsUnavailable(): Unit = status.set(Down)
28

```

test/scala/akka/cluster/HeartbeatNodeRingSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: bb
Enclosing Method: HeartbeatNodeRingSpec()
File: test/scala/akka/cluster/HeartbeatNodeRingSpec.scala:21
Taint Flags:

```

18 val ee = UniqueAddress(Address("akka", "sys", "ee", 2552), 5L)
19 val ff = UniqueAddress(Address("akka", "sys", "ff", 2552), 6L)
20
21 val nodes = Set(aa, bb, cc, dd, ee, ff)
22
23 "A HashedNodeRing" must {
24

```

test/scala/akka/cluster/HeartbeatNodeRingSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: dd
Enclosing Method: HeartbeatNodeRingSpec()
File: test/scala/akka/cluster/HeartbeatNodeRingSpec.scala:21
Taint Flags:

```

18 val ee = UniqueAddress(Address("akka", "sys", "ee", 2552), 5L)
19 val ff = UniqueAddress(Address("akka", "sys", "ff", 2552), 6L)
20
21 val nodes = Set(aa, bb, cc, dd, ee, ff)
22
23 "A HashedNodeRing" must {
24

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/DowningProviderSpec.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function)	Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: actorPropsAccessed
Enclosing Method: DummyDowningProvider()
File: test/scala/akka/cluster/DowningProviderSpec.scala:33
Taint Flags:

```

30
31 val actorPropsAccessed = new AtomicBoolean(false)
32 override val downingActorProps: Option[Props] = {
33 actorPropsAccessed.set(true)
34 None
35 }
36 }
```

main/scala/akka/cluster/ClusterDaemon.scala, line 338 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: selfUniqueAddress
Enclosing Method: ClusterCoreDaemon()
File: main/scala/akka/cluster/ClusterDaemon.scala:338
Taint Flags:

```

335
336 protected def selfUniqueAddress = cluster.selfUniqueAddress
337
338 val vclockNode = VectorClock.Node(Gossip.vclockName(selfUniqueAddress))
339 val gossipTargetSelector = new GossipTargetSelector(
340 ReduceGossipDifferentViewProbability,
341 cluster.settings.MultiDataCenter.CrossDcGossipProbability)
```

main/scala/akka/cluster/Cluster.scala, line 72 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/Cluster.scala, line 72 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details**

Sink: FunctionCall: settings
Enclosing Method: Cluster()
File: main/scala/akka/cluster/Cluster.scala:72
Taint Flags:

```
69 import ClusterLogger._
70 import settings._
71
72 private val joinConfigCompatChecker: JoinConfigCompatChecker = JoinConfigCompatChecker.load(system, settings)
73
74 /**
75  * The address including a `uid` of this cluster member.
```

main/scala/akka/cluster/Cluster.scala, line 154 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: settings
Enclosing Method: Cluster()
File: main/scala/akka/cluster/Cluster.scala:154
Taint Flags:

```
151 * INTERNAL API
152 */
153 private[cluster] val scheduler: Scheduler = {
154   if (system.scheduler.maxFrequency < 1.second / SchedulerTickDuration) {
155     logInfo(
156       "Using a dedicated scheduler for cluster. Default scheduler can be used if configured " +
157       "with 'akka.scheduler.tick-duration' [{ } ms] <= 'akka.cluster.scheduler.tick-duration' [{ } ms].",
```

main/scala/akka/cluster/Cluster.scala, line 155 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: settings
Enclosing Method: Cluster()



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/Cluster.scala, line 155 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** main/scala/akka/cluster/Cluster.scala:155**Taint Flags:**

```
152 */
153 private[cluster] val scheduler: Scheduler = {
154   if (system.scheduler.maxFrequency < 1.second / SchedulerTickDuration) {
155     logInfo(
156       "Using a dedicated scheduler for cluster. Default scheduler can be used if configured " +
157       "'with 'akka.scheduler.tick-duration' [{ } ms] <= 'akka.cluster.scheduler.tick-duration' [{ } ms].",
158       (1000 / system.scheduler.maxFrequency).toInt,
```

main/scala/akka/cluster/Cluster.scala, line 161 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: settings**Enclosing Method:** Cluster()**File:** main/scala/akka/cluster/Cluster.scala:161**Taint Flags:**

```
158 (1000 / system.scheduler.maxFrequency).toInt,
159 SchedulerTickDuration.toMillis)
160
161 val cfg = ConfigFactory
162   .parseString(s"akka.scheduler.tick-duration=${SchedulerTickDuration.toMillis}ms")
163   .withFallback(system.settings.config)
164 val threadFactory = system.threadFactory match {
```

main/scala/akka/cluster/Cluster.scala, line 195 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: settings**Enclosing Method:** Cluster()**File:** main/scala/akka/cluster/Cluster.scala:195**Taint Flags:**

```
192 }
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/Cluster.scala, line 195 (Code Correctness: Constructor Invokes Overridable Function)	Low

```

193
194 // create supervisor for daemons under path "/system/cluster"
195 private val clusterDaemons: ActorRef = {
196   system.systemActorOf(
197     Props(classOf[ClusterDaemon], joinConfigCompatChecker).withDispatcher(UseDispatcher).withDeploy(Deploy.local),
198     name = "cluster")

```

main/scala/akka/cluster/Cluster.scala, line 223 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: settings
Enclosing Method: Cluster()
File: main/scala/akka/cluster/Cluster.scala:223
Taint Flags:

```

220
221 system.registerOnTermination(shutdown())
222
223 if (JmxEnabled)
224   clusterJmx = {
225     val jmx = new ClusterJmx(this, log)
226     jmx.createMBean()

```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: CrossDcHeartbeatSender()
File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:44
Taint Flags:

```

41
42 val cluster = Cluster(context.system)
43
44 val verboseHeartbeat = cluster.settings.Debug.VerboseHeartbeatLogging

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 45 import cluster.{ scheduler, selfAddress, selfDataCenter, selfUniqueAddress } 46 import cluster.settings.PeriodicTasksInitialDelay 47 import context.dispatcher </pre>	
main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 50 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cluster Enclosing Method: CrossDcHeartbeatSender() File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:50 Taint Flags:	
<pre> 47 import context.dispatcher 48 49 private val clusterLogger = 50 new cluster.ClusterLogger(51 Logging.withMarker(context.system, ActorWithLogClass(this, ClusterLogClass.ClusterHeartbeat))) 52 import clusterLogger._ 53 </pre>	
main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 60 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cluster Enclosing Method: CrossDcHeartbeatSender() File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:60 Taint Flags:	
<pre> 57 val isExternalClusterMember: Member => Boolean = 58 member => member.dataCenter != cluster.selfDataCenter 59 60 val crossDcSettings: cluster.settings.CrossDcFailureDetectorSettings = 61 cluster.settings.MultiDataCenter.CrossDcFailureDetectorSettings 62 63 val crossDcFailureDetector = cluster.crossDcFailureDetector </pre>	



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 63 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** CrossDcHeartbeatSender()**File:** main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:63**Taint Flags:**

```
60 val crossDcSettings: cluster.settings.CrossDcFailureDetectorSettings =  
61 cluster.settings.MultiDataCenter.CrossDcFailureDetectorSettings  
62  
63 val crossDcFailureDetector = cluster.crossDcFailureDetector  
64  
65 var sequenceNr: Long = 0  
66
```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 72 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** CrossDcHeartbeatSender()**File:** main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:72**Taint Flags:**

```
69 ClusterHeartbeatSender.Heartbeat(selfAddress, sequenceNr, System.nanoTime())  
70 }  
71  
72 var dataCentersState: CrossDcHeartbeatingState = CrossDcHeartbeatingState.init(  
73 selfDataCenter,  
74 crossDcFailureDetector,  
75 crossDcSettings.NrOfMonitoringActors,
```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** CrossDcHeartbeatSender()**File:** main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:79**Taint Flags:**

```
76 immutable.SortedSet.empty)
77
78 // start periodic heartbeat to other nodes in cluster
79 val heartbeatTask = scheduler.scheduleWithFixedDelay(
80 PeriodicTasksInitialDelay max crossDcSettings.HeartbeatInterval,
81 crossDcSettings.HeartbeatInterval,
82 self,
```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** CrossDcHeartbeatSender()**File:** main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:79**Taint Flags:**

```
76 immutable.SortedSet.empty)
77
78 // start periodic heartbeat to other nodes in cluster
79 val heartbeatTask = scheduler.scheduleWithFixedDelay(
80 PeriodicTasksInitialDelay max crossDcSettings.HeartbeatInterval,
81 crossDcSettings.HeartbeatInterval,
82 self,
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: middleNode**Enclosing Method:** VectorClockPerfSpec()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/VectorClockPerfSpec.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/VectorClockPerfSpec.scala:46

Taint Flags:

```

43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
48 val vcConcurrentMiddle = vcBaseMiddle :+ middleNode
49

```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: middleNode

Enclosing Method: VectorClockPerfSpec()

File: test/scala/akka/cluster/VectorClockPerfSpec.scala:48

Taint Flags:

```

45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
48 val vcConcurrentMiddle = vcBaseMiddle :+ middleNode
49
50 def checkThunkFor(vc1: VectorClock, vc2: VectorClock, thunk: (VectorClock, VectorClock) => Unit, times: Int): Unit = {
51 val vcc1 = copyVectorClock(vc1)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 66 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: state

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:66

Taint Flags:

```

63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)

```



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

Package: akka.cluster

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 66 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```

64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)

```

main/scala/akka/cluster/Gossip.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: \$default\$4
Enclosing Method: Gossip()
File: main/scala/akka/cluster/Gossip.scala:23
Taint Flags:

```

20 private[cluster] object Gossip {
21   type Timestamp = Long
22   val emptyMembers: immutable.SortedSet[Member] = immutable.SortedSet.empty
23   val empty: Gossip = new Gossip(Gossip.emptyMembers)
24
25   def apply(members: immutable.SortedSet[Member]): Gossip =
26     if (members.isEmpty) empty else empty.copy(members = members)

```

main/scala/akka/cluster/Cluster.scala, line 224 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: clusterJmx_
Enclosing Method: Cluster()
File: main/scala/akka/cluster/Cluster.scala:224
Taint Flags:

```

221 system.registerOnTermination(shutdown())
222
223 if (JmxEnabled)
224   clusterJmx = {

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/Cluster.scala, line 224 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 225 val jmx = new ClusterJmx(this, log) 226 jmx.createMBean() 227 Some(jmx) </pre>	
test/scala/akka/cluster/GossipSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: a1 Enclosing Method: GossipSpec() File: test/scala/akka/cluster/GossipSpec.scala:21 Taint Flags:	
<pre> 18 import MemberStatus._ 19 20 val a1 = TestMember(Address("akka", "sys", "a", 2552), Up) 21 val a2 = TestMember(a1.address, Joining) 22 val b1 = TestMember(Address("akka", "sys", "b", 2552), Up) 23 val b2 = TestMember(b1.address, Removed) 24 val c1 = TestMember(Address("akka", "sys", "c", 2552), Leaving) </pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 369 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: coordShutdown Enclosing Method: ClusterCoreDaemon() File: main/scala/akka/cluster/ClusterDaemon.scala:369 Taint Flags:	
<pre> 366 var exitingTasksInProgress = false 367 val selfExiting = Promise[Done]() 368 val coordShutdown = CoordinatedShutdown(context.system) 369 coordShutdown.addTask(CoordinatedShutdown.PhaseClusterExiting, "wait-exiting") { () => 370 if (latestGossip.members.isEmpty) 371 Future.successful(Done) // not joined yet 372 else </pre>	



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

Package: akka.cluster

main/scala/akka/cluster/ClusterDaemon.scala, line 375 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: coordShutdown
Enclosing Method: ClusterCoreDaemon()
File: main/scala/akka/cluster/ClusterDaemon.scala:375
Taint Flags:

```

372 else
373 selfExiting.future
374 }
375 coordShutdown.addTask(CoordinatedShutdown.PhaseClusterExitingDone, "exiting-completed") {
376 val sys = context.system
377 () =>
378 if (Cluster(sys).isTerminated || Cluster(sys).selfMember.status == Down)

```

main/scala/akka/cluster/Cluster.scala, line 204 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: liftedTree1
Enclosing Method: Cluster()
File: main/scala/akka/cluster/Cluster.scala:204
Taint Flags:

```

201 /**
202 * INTERNAL API
203 */
204 private[cluster] val clusterCore: ActorRef = {
205 implicit val timeout = system.settings.CreationTimeout
206 try {
207 Await.result((clusterDaemons ? InternalClusterAction.GetClusterCoreRef).mapTo[ActorRef], timeout.duration)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 91 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 91 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: g9**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:91**Taint Flags:**

```
88 val g9 = Gossip(  
89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),  
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress)))  
91 val state9 = state(g9, aUp.uniqueAddress)  
92 val g10 = Gossip(  
93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),  
94 overview = GossipOverview(reachability = Reachability.empty))
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aExiting**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:48**Taint Flags:**

```
45 val aUp = TestMember(Address(protocol, "sys", "a", 2552), Up)  
46 val aLeaving = aUp.copy(status = Leaving)  
47 val aExiting = aLeaving.copy(status = Exiting)  
48 val aRemoved = aExiting.copy(status = Removed)  
49 val bExiting = TestMember(Address(protocol, "sys", "b", 2552), Exiting)  
50 val bRemoved = bExiting.copy(status = Removed)  
51 val cJoining = TestMember(Address(protocol, "sys", "c", 2552), Joining, Set("GRP"))
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 81 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aExiting**Enclosing Method:** ClusterDomainEventPublisherSpec()

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 81 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:81**Taint Flags:**

```
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(
84 members = SortedSet(aUp, bExiting, cUp, dUp),
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 47 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aLeaving**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:47**Taint Flags:**

```
44
45 val aUp = TestMember(Address(protocol, "sys", "a", 2552), Up)
46 val aLeaving = aUp.copy(status = Leaving)
47 val aExiting = aLeaving.copy(status = Exiting)
48 val aRemoved = aExiting.copy(status = Removed)
49 val bExiting = TestMember(Address(protocol, "sys", "b", 2552), Exiting)
50 val bRemoved = bExiting.copy(status = Removed)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aLeaving**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:79**Taint Flags:**

```
76 .seen(cUp.uniqueAddress)
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
77 .seen(a51Up.uniqueAddress)
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
```

test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: reachability1**Enclosing Method:** ReachabilityPerfSpec()**File:** test/scala/akka/cluster/ReachabilityPerfSpec.scala:42**Taint Flags:**

```
39 }
40
41 val reachability1 = createReachabilityOfSize(Reachability.empty, nodesSize)
42 val reachability2 = createReachabilityOfSize(reachability1, nodesSize)
43 val reachability3 = addUnreachable(reachability1, nodesSize / 2)
44 val allowed = reachability1.versions.keySet
45
```

test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: reachability1**Enclosing Method:** ReachabilityPerfSpec()**File:** test/scala/akka/cluster/ReachabilityPerfSpec.scala:43**Taint Flags:**

```
40
41 val reachability1 = createReachabilityOfSize(Reachability.empty, nodesSize)
42 val reachability2 = createReachabilityOfSize(reachability1, nodesSize)
43 val reachability3 = addUnreachable(reachability1, nodesSize / 2)
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 44 val allowed = reachability1.versions.keySet 45 46 private def checkThunkFor(</pre>	
test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: reachability1 Enclosing Method: ReachabilityPerfSpec() File: test/scala/akka/cluster/ReachabilityPerfSpec.scala:44 Taint Flags:	
<pre> 41 val reachability1 = createReachabilityOfSize(Reachability.empty, nodesSize) 42 val reachability2 = createReachabilityOfSize(reachability1, nodesSize) 43 val reachability3 = addUnreachable(reachability1, nodesSize / 2) 44 val allowed = reachability1.versions.keySet 45 46 private def checkThunkFor(47 r1: Reachability, </pre>	
main/scala/akka/cluster/Gossip.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: \$default\$2 Enclosing Method: Gossip() File: main/scala/akka/cluster/Gossip.scala:23 Taint Flags:	
<pre> 20 private[cluster] object Gossip { 21 type Timestamp = Long 22 val emptyMembers: immutable.SortedSet[Member] = immutable.SortedSet.empty 23 val empty: Gossip = new Gossip(Gossip.emptyMembers) 24 25 def apply(members: immutable.SortedSet[Member]): Gossip = 26 if (members.isEmpty) empty else empty.copy(members = members) </pre>	



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

Package: akka.cluster

main/scala/akka/cluster/ClusterEvent.scala, line 680 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: emptyMembershipState
Enclosing Method: ClusterDomainEventPublisher()
File: main/scala/akka/cluster/ClusterEvent.scala:680
Taint Flags:

```

677 cluster.selfUniqueAddress,
678 cluster.settings.SelfDataCenter,
679 cluster.settings.MultiDataCenter.CrossDcConnections)
680 var membershipState: MembershipState = emptyMembershipState
681 def selfDc = cluster.settings.SelfDataCenter
682
683 override def preRestart(reason: Throwable, message: Option[Any]): Unit = {

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 72 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: state
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:72
Taint Flags:

```

69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)
75 .seen(bExiting.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 27 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 27 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: cRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:27**Taint Flags:**

```
24 val bDown = TestMember(Address("akka", "sys", "b", 2552), Down, bRoles)
25 val bRemoved = TestMember(Address("akka", "sys", "b", 2552), Removed, bRoles)
26 val cRoles = Set.empty[String]
27 val cUp = TestMember(Address("akka", "sys", "c", 2552), Up, cRoles)
28 val cLeaving = TestMember(Address("akka", "sys", "c", 2552), Leaving, cRoles)
29 val dRoles = Set("DD", "DE")
30 val dLeaving = TestMember(Address("akka", "sys", "d", 2552), Leaving, dRoles)
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 28 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:28**Taint Flags:**

```
25 val bRemoved = TestMember(Address("akka", "sys", "b", 2552), Removed, bRoles)
26 val cRoles = Set.empty[String]
27 val cUp = TestMember(Address("akka", "sys", "c", 2552), Up, cRoles)
28 val cLeaving = TestMember(Address("akka", "sys", "c", 2552), Leaving, cRoles)
29 val dRoles = Set("DD", "DE")
30 val dLeaving = TestMember(Address("akka", "sys", "d", 2552), Leaving, dRoles)
31 val dExiting = TestMember(Address("akka", "sys", "d", 2552), Exiting, dRoles)
```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 104 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** ClusterHeartbeatSender()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterHeartbeat.scala, line 104 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: main/scala/akka/cluster/ClusterHeartbeat.scala:104

Taint Flags:

```

101 import ClusterHeartbeatSender._
102
103 val cluster = Cluster(context.system)
104 val verboseHeartbeat = cluster.settings.Debug.VerboseHeartbeatLogging
105 import cluster.scheduler
106 import cluster.selfAddress
107 import cluster.selfUniqueAddress

```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 112 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster

Enclosing Method: ClusterHeartbeatSender()

File: main/scala/akka/cluster/ClusterHeartbeat.scala:112

Taint Flags:

```

109 import context.dispatcher
110
111 private val clusterLogger =
112 new cluster.ClusterLogger(
113 Logging.withMarker(context.system, ActorWithLogClass(this, ClusterLogClass.ClusterHeartbeat)))
114 import clusterLogger._
115

```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 126 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster

Enclosing Method: ClusterHeartbeatSender()

File: main/scala/akka/cluster/ClusterHeartbeat.scala:126

Taint Flags:

```

123 Heartbeat(selfAddress, sequenceNr, System.nanoTime())

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterHeartbeat.scala, line 126 (Code Correctness: Constructor Invokes Overridable Function)	Low

```

124 }
125
126 val failureDetector = cluster.failureDetector
127
128 var state: ClusterHeartbeatSenderState = ClusterHeartbeatSenderState(
129 ring = HeartbeatNodeRing(selfUniqueAddress, Set(selfUniqueAddress), Set.empty, MonitoredByNrOfMembers),

```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 129 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterHeartbeatSender()
File: main/scala/akka/cluster/ClusterHeartbeat.scala:129
Taint Flags:

```

126 val failureDetector = cluster.failureDetector
127
128 var state: ClusterHeartbeatSenderState = ClusterHeartbeatSenderState(
129 ring = HeartbeatNodeRing(selfUniqueAddress, Set(selfUniqueAddress), Set.empty, MonitoredByNrOfMembers),
130 oldReceiversNowUnreachable = Set.empty[UniqueAddress],
131 failureDetector)
132

```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 129 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterHeartbeatSender()
File: main/scala/akka/cluster/ClusterHeartbeat.scala:129
Taint Flags:

```

126 val failureDetector = cluster.failureDetector
127
128 var state: ClusterHeartbeatSenderState = ClusterHeartbeatSenderState(
129 ring = HeartbeatNodeRing(selfUniqueAddress, Set(selfUniqueAddress), Set.empty, MonitoredByNrOfMembers),

```



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

Package: akka.cluster

main/scala/akka/cluster/ClusterHeartbeat.scala, line 129 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```
130 oldReceiversNowUnreachable = Set.empty[UniqueAddress],
131 failureDetector)
132
```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 129 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster

Enclosing Method: ClusterHeartbeatSender()

File: main/scala/akka/cluster/ClusterHeartbeat.scala:129

Taint Flags:

```
126 val failureDetector = cluster.failureDetector
127
128 var state: ClusterHeartbeatSenderState = ClusterHeartbeatSenderState(
129 ring = HeartbeatNodeRing(selfUniqueAddress, Set(selfUniqueAddress), Set.empty, MonitoredByNrOfMembers),
130 oldReceiversNowUnreachable = Set.empty[UniqueAddress],
131 failureDetector)
132
```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 134 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster

Enclosing Method: ClusterHeartbeatSender()

File: main/scala/akka/cluster/ClusterHeartbeat.scala:134

Taint Flags:

```
131 failureDetector)
132
133 // start periodic heartbeat to other nodes in cluster
134 val heartbeatTask =
135 scheduler.scheduleWithFixedDelay(
136 PeriodicTasksInitialDelay max HeartbeatInterval,
137 HeartbeatInterval,
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterHeartbeat.scala, line 134 (Code Correctness: Constructor Invokes Overridable Function)	Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterHeartbeatSender()
File: main/scala/akka/cluster/ClusterHeartbeat.scala:134
Taint Flags:

```

131 failureDetector)
132
133 // start periodic heartbeat to other nodes in cluster
134 val heartbeatTask =
135 scheduler.scheduleWithFixedDelay(
136 PeriodicTasksInitialDelay max HeartbeatInterval,
137 HeartbeatInterval,
```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 134 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterHeartbeatSender()
File: main/scala/akka/cluster/ClusterHeartbeat.scala:134
Taint Flags:

```

131 failureDetector)
132
133 // start periodic heartbeat to other nodes in cluster
134 val heartbeatTask =
135 scheduler.scheduleWithFixedDelay(
136 PeriodicTasksInitialDelay max HeartbeatInterval,
137 HeartbeatInterval,
```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 142 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterHeartbeat.scala, line 142 (Code Correctness: Constructor Invokes Overridable Function)	Low

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterHeartbeatSender()
File: main/scala/akka/cluster/ClusterHeartbeat.scala:142
Taint Flags:

```

139 HeartbeatTick)
140
141 // used for logging warning if actual tick interval is unexpected (e.g. due to starvation)
142 private var tickTimestamp = System.nanoTime() + (PeriodicTasksInitialDelay max HeartbeatInterval).toNanos
143
144 override def preStart(): Unit = {
145 cluster.subscribe(self, classOf[MemberEvent], classOf[ReachabilityEvent])

```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 142 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterHeartbeatSender()
File: main/scala/akka/cluster/ClusterHeartbeat.scala:142
Taint Flags:

```

139 HeartbeatTick)
140
141 // used for logging warning if actual tick interval is unexpected (e.g. due to starvation)
142 private var tickTimestamp = System.nanoTime() + (PeriodicTasksInitialDelay max HeartbeatInterval).toNanos
143
144 override def preStart(): Unit = {
145 cluster.subscribe(self, classOf[MemberEvent], classOf[ReachabilityEvent])

```

main/scala/akka/cluster/Gossip.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: \$default\$3
Enclosing Method: Gossip()



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/Gossip.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** main/scala/akka/cluster/Gossip.scala:23**Taint Flags:**

```
20 private[cluster] object Gossip {  
21   type Timestamp = Long  
22   val emptyMembers: immutable.SortedSet[Member] = immutable.SortedSet.empty  
23   val empty: Gossip = new Gossip(Gossip.emptyMembers)  
24  
25   def apply(members: immutable.SortedSet[Member]): Gossip =  
26   if (members.isEmpty) empty else empty.copy(members = members)
```

main/scala/akka/cluster/ClusterSettings.scala, line 41 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:41**Taint Flags:**

```
38 import ClusterSettings._  
39 private val cc = config.getConfig("akka.cluster")  
40  
41 val LogInfoVerbose: Boolean = cc.getBoolean("log-info-verbose")  
42 val LogInfo: Boolean = LogInfoVerbose || cc.getBoolean("log-info")  
43 val FailureDetectorConfig: Config = cc.getConfig("failure-detector")  
44 val FailureDetectorImplementationClass: String = FailureDetectorConfig.getString("implementation-class")
```

main/scala/akka/cluster/ClusterSettings.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:42**Taint Flags:**

```
39 private val cc = config.getConfig("akka.cluster")
```



Code Correctness: Constructor Invokes Overridable Function		Low
Package: akka.cluster		
main/scala/akka/cluster/ClusterSettings.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)		Low
<pre> 40 41 val LogInfoVerbose: Boolean = cc.getBoolean("log-info-verbose") 42 val LogInfo: Boolean = LogInfoVerbose cc.getBoolean("log-info") 43 val FailureDetectorConfig: Config = cc.getConfig("failure-detector") 44 val FailureDetectorImplementationClass: String = FailureDetectorConfig.getString("implementation-class") 45 val HeartbeatInterval: FiniteDuration = { </pre>		
main/scala/akka/cluster/ClusterSettings.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function)		Low
Issue Details Kingdom: Code Quality Scan Engine: SCA (Structural)		
Sink Details Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:43 Taint Flags:		
<pre> 40 41 val LogInfoVerbose: Boolean = cc.getBoolean("log-info-verbose") 42 val LogInfo: Boolean = LogInfoVerbose cc.getBoolean("log-info") 43 val FailureDetectorConfig: Config = cc.getConfig("failure-detector") 44 val FailureDetectorImplementationClass: String = FailureDetectorConfig.getString("implementation-class") 45 val HeartbeatInterval: FiniteDuration = { 46 FailureDetectorConfig.getMillisDuration("heartbeat-interval") </pre>		
main/scala/akka/cluster/ClusterSettings.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)		Low
Issue Details Kingdom: Code Quality Scan Engine: SCA (Structural)		
Sink Details Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:79 Taint Flags:		
<pre> 76 new CrossDcFailureDetectorSettings(cc.getConfig("multi-data-center.failure-detector")) 77 } 78 79 val SeedNodes: immutable.IndexedSeq[Address] = </pre>		



Code Correctness: Constructor Invokes Overridable Function		Low
Package: akka.cluster		
main/scala/akka/cluster/ClusterSettings.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)		Low
<pre> 80 immutableSeq(cc.getStringList("seed-nodes")).map { 81 case AddressFromURIStrng(address) => address 82 case _ => throw new RuntimeException() // compiler exhaustiveness check pleaser </pre>		
main/scala/akka/cluster/ClusterSettings.scala, line 84 (Code Correctness: Constructor Invokes Overridable Function)		Low
Issue Details		
Kingdom: Code Quality Scan Engine: SCA (Structural)		
Sink Details		
Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:84 Taint Flags:		
<pre> 81 case AddressFromURIStrng(address) => address 82 case _ => throw new RuntimeException() // compiler exhaustiveness check pleaser 83 }.toVector 84 val SeedNodeTimeout: FiniteDuration = cc.getMillisDuration("seed-node-timeout") 85 val RetryUnsuccessfulJoinAfter: Duration = { 86 val key = "retry-unsuccessful-join-after" 87 toRootLowerCase(cc.getString(key)) match { </pre>		
main/scala/akka/cluster/ClusterSettings.scala, line 87 (Code Correctness: Constructor Invokes Overridable Function)		Low
Issue Details		
Kingdom: Code Quality Scan Engine: SCA (Structural)		
Sink Details		
Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:87 Taint Flags:		
<pre> 84 val SeedNodeTimeout: FiniteDuration = cc.getMillisDuration("seed-node-timeout") 85 val RetryUnsuccessfulJoinAfter: Duration = { 86 val key = "retry-unsuccessful-join-after" 87 toRootLowerCase(cc.getString(key)) match { 88 case "off" => Duration.Undefined 89 case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off") 90 } </pre>		

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 89 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:89**Taint Flags:**

```
86 val key = "retry-unsuccessful-join-after"
87 toRootLowerCase(cc.getString(key)) match {
88 case "off" => Duration.Undefined
89 case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")
90 }
91 }
92 val ShutdownAfterUnsuccessfulJoinSeedNodes: Duration = {
```

main/scala/akka/cluster/ClusterSettings.scala, line 94 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:94**Taint Flags:**

```
91 }
92 val ShutdownAfterUnsuccessfulJoinSeedNodes: Duration = {
93 val key = "shutdown-after-unsuccessful-join-seed-nodes"
94 toRootLowerCase(cc.getString(key)) match {
95 case "off" => Duration.Undefined
96 case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")
97 }
```

main/scala/akka/cluster/ClusterSettings.scala, line 96 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 96 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:96**Taint Flags:**

```
93 val key = "shutdown-after-unsuccessful-join-seed-nodes"
94 toRootLowerCase(cc.getString(key)) match {
95   case "off" => Duration.Undefined
96   case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")
97 }
98 }
99 val PeriodicTasksInitialDelay: FiniteDuration = cc.getMillisDuration("periodic-tasks-initial-delay")
```

main/scala/akka/cluster/ClusterSettings.scala, line 99 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:99**Taint Flags:**

```
96 case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")
97 }
98 }
99 val PeriodicTasksInitialDelay: FiniteDuration = cc.getMillisDuration("periodic-tasks-initial-delay")
100 val GossipInterval: FiniteDuration = cc.getMillisDuration("gossip-interval")
101 val GossipTimeToLive: FiniteDuration = {
102   cc.getMillisDuration("gossip-time-to-live")
```

main/scala/akka/cluster/ClusterSettings.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterSettings.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: main/scala/akka/cluster/ClusterSettings.scala:100

Taint Flags:

```

97 }
98 }
99 val PeriodicTasksInitialDelay: FiniteDuration = cc.getMillisDuration("periodic-tasks-initial-delay")
100 val GossipInterval: FiniteDuration = cc.getMillisDuration("gossip-interval")
101 val GossipTimeToLive: FiniteDuration = {
102   cc.getMillisDuration("gossip-time-to-live")
103 }.requiring(_ > Duration.Zero, "gossip-time-to-live must be > 0")

```

main/scala/akka/cluster/ClusterSettings.scala, line 101 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc

Enclosing Method: ClusterSettings()

File: main/scala/akka/cluster/ClusterSettings.scala:101

Taint Flags:

```

98 }
99 val PeriodicTasksInitialDelay: FiniteDuration = cc.getMillisDuration("periodic-tasks-initial-delay")
100 val GossipInterval: FiniteDuration = cc.getMillisDuration("gossip-interval")
101 val GossipTimeToLive: FiniteDuration = {
102   cc.getMillisDuration("gossip-time-to-live")
103 }.requiring(_ > Duration.Zero, "gossip-time-to-live must be > 0")
104 val LeaderActionsInterval: FiniteDuration = cc.getMillisDuration("leader-actions-interval")

```

main/scala/akka/cluster/ClusterSettings.scala, line 104 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc

Enclosing Method: ClusterSettings()

File: main/scala/akka/cluster/ClusterSettings.scala:104

Taint Flags:

```

101 val GossipTimeToLive: FiniteDuration = {

```



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

Package: akka.cluster

main/scala/akka/cluster/ClusterSettings.scala, line 104 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

```

102 cc.getMillisDuration("gossip-time-to-live")
103 }.requiring(_ > Duration.Zero, "gossip-time-to-live must be > 0")
104 val LeaderActionsInterval: FiniteDuration = cc.getMillisDuration("leader-actions-interval")
105 val UnreachableNodesReaperInterval: FiniteDuration = cc.getMillisDuration("unreachable-nodes-reaper-interval")
106 val PublishStatsInterval: Duration = {
107 val key = "publish-stats-interval"

```

main/scala/akka/cluster/ClusterSettings.scala, line 105 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:105
Taint Flags:

```

102 cc.getMillisDuration("gossip-time-to-live")
103 }.requiring(_ > Duration.Zero, "gossip-time-to-live must be > 0")
104 val LeaderActionsInterval: FiniteDuration = cc.getMillisDuration("leader-actions-interval")
105 val UnreachableNodesReaperInterval: FiniteDuration = cc.getMillisDuration("unreachable-nodes-reaper-interval")
106 val PublishStatsInterval: Duration = {
107 val key = "publish-stats-interval"
108 toRootLowerCase(cc.getString(key)) match {

```

main/scala/akka/cluster/ClusterSettings.scala, line 108 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:108
Taint Flags:

```

105 val UnreachableNodesReaperInterval: FiniteDuration = cc.getMillisDuration("unreachable-nodes-reaper-interval")
106 val PublishStatsInterval: Duration = {
107 val key = "publish-stats-interval"
108 toRootLowerCase(cc.getString(key)) match {

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterSettings.scala, line 108 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 109 case "off" => Duration.Undefined 110 case _ => cc.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off") 111 } </pre>	
main/scala/akka/cluster/ClusterSettings.scala, line 110 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:110 Taint Flags:	
<pre> 107 val key = "publish-stats-interval" 108 toRootLowerCase(cc.getString(key)) match { 109 case "off" => Duration.Undefined 110 case _ => cc.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off") 111 } 112 } 113 </pre>	
main/scala/akka/cluster/ClusterSettings.scala, line 117 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:117 Taint Flags:	
<pre> 114 /** 115 * Is in fact always a `FiniteDuration` but needs to stay `Duration` for binary compatibility 116 */ 117 val PruneGossipTombstonesAfter: Duration = { 118 val key = "prune-gossip-tombstones-after" 119 cc.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s") 120 } </pre>	



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

Package: akka.cluster

main/scala/akka/cluster/ClusterSettings.scala, line 124 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:124
Taint Flags:

```

121
122 val DownRemovalMargin: FiniteDuration = {
123   val key = "down-removal-margin"
124   toRootLowerCase(cc.getString(key)) match {
125     case "off" => Duration.Zero
126     case _ => cc.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off")
127   }

```

main/scala/akka/cluster/ClusterSettings.scala, line 126 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:126
Taint Flags:

```

123   val key = "down-removal-margin"
124   toRootLowerCase(cc.getString(key)) match {
125     case "off" => Duration.Zero
126     case _ => cc.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off")
127   }
128 }
129

```

main/scala/akka/cluster/ClusterSettings.scala, line 131 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 131 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details**

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:131
Taint Flags:

```
128 }  
129  
130 val DowningProviderClassName: String = {  
131 val name = cc.getString("downing-provider-class")  
132 if (name.nonEmpty) name  
133 else classOf[NoDowning].getName  
134 }
```

main/scala/akka/cluster/ClusterSettings.scala, line 136 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:136
Taint Flags:

```
133 else classOf[NoDowning].getName  
134 }  
135  
136 val QuarantineRemovedNodeAfter: FiniteDuration =  
137 cc.getMillisDuration("quarantine-removed-node-after")  
138 .requiring(_ > Duration.Zero, "quarantine-removed-node-after must be > 0")  
139
```

main/scala/akka/cluster/ClusterSettings.scala, line 142 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc
Enclosing Method: ClusterSettings()



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 142 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** main/scala/akka/cluster/ClusterSettings.scala:142**Taint Flags:**

```
139
140 val WeaklyUpAfter: FiniteDuration = {
141   val key = "allow-weakly-up-members"
142   toRootLowerCase(cc.getString(key)) match {
143     case "off" => Duration.Zero
144     case "on" => 7.seconds // for backwards compatibility when it wasn't a duration
145     case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")
```

main/scala/akka/cluster/ClusterSettings.scala, line 145 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:145**Taint Flags:**

```
142 toRootLowerCase(cc.getString(key)) match {
143   case "off" => Duration.Zero
144   case "on" => 7.seconds // for backwards compatibility when it wasn't a duration
145   case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")
146 }
147 }
148
```

main/scala/akka/cluster/ClusterSettings.scala, line 151 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:151**Taint Flags:**

```
148
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 151 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
149 val AllowWeaklyUpMembers: Boolean = WeaklyUpAfter != Duration.Zero
150
151 val SelfDataCenter: DataCenter = cc.getString("multi-data-center.self-data-center")
152
153 val Roles: Set[String] = {
154 val configuredRoles = immutableSeq(cc.getStringList("roles")).toSet.requiring(
```

main/scala/akka/cluster/ClusterSettings.scala, line 154 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:154
Taint Flags:

```
151 val SelfDataCenter: DataCenter = cc.getString("multi-data-center.self-data-center")
152
153 val Roles: Set[String] = {
154 val configuredRoles = immutableSeq(cc.getStringList("roles")).toSet.requiring(
155   _.forall(!__.startsWith(DcRolePrefix)),
156   s"Roles must not start with '$DcRolePrefix' as that is reserved for the cluster self-data-center setting")
157
```

main/scala/akka/cluster/ClusterSettings.scala, line 161 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:161
Taint Flags:

```
158 configuredRoles + s"$DcRolePrefix$SelfDataCenter"
159 }
160
161 val AppVersion: Version =
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterSettings.scala, line 161 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 162 Version(cc.getString("app-version")) 163 164 val MinNrOfMembers: Int = { </pre>	
main/scala/akka/cluster/ClusterSettings.scala, line 164 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:164 Taint Flags:	
<pre> 161 val AppVersion: Version = 162 Version(cc.getString("app-version")) 163 164 val MinNrOfMembers: Int = { 165 cc.getInt("min-nr-of-members") 166 }.requiring(_ > 0, "min-nr-of-members must be > 0") 167 val MinNrOfMembersOfRole: Map[String, Int] = { </pre>	
main/scala/akka/cluster/ClusterSettings.scala, line 167 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:167 Taint Flags:	
<pre> 164 val MinNrOfMembers: Int = { 165 cc.getInt("min-nr-of-members") 166 }.requiring(_ > 0, "min-nr-of-members must be > 0") 167 val MinNrOfMembersOfRole: Map[String, Int] = { 168 import akka.util.compat.JavaConverters._ 169 cc.getConfig("role") 170 .root </pre>	



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 177 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:177**Taint Flags:**

```
174 }  
175 .toMap  
176 }  
177 val RunCoordinatedShutdownWhenDown: Boolean = cc.getBoolean("run-coordinated-shutdown-when-down")  
178 val JmxEnabled: Boolean = cc.getBoolean("jmx.enabled")  
179 val JmxMultiMbeansInSameEnabled: Boolean = cc.getBoolean("jmx.multi-mbeans-in-same-jvm")  
180 val UseDispatcher: String = cc.getString("use-dispatcher")
```

main/scala/akka/cluster/ClusterSettings.scala, line 178 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:178**Taint Flags:**

```
175 .toMap  
176 }  
177 val RunCoordinatedShutdownWhenDown: Boolean = cc.getBoolean("run-coordinated-shutdown-when-down")  
178 val JmxEnabled: Boolean = cc.getBoolean("jmx.enabled")  
179 val JmxMultiMbeansInSameEnabled: Boolean = cc.getBoolean("jmx.multi-mbeans-in-same-jvm")  
180 val UseDispatcher: String = cc.getString("use-dispatcher")  
181 val GossipDifferentViewProbability: Double = cc.getDouble("gossip-different-view-probability")
```

main/scala/akka/cluster/ClusterSettings.scala, line 179 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 179 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:179**Taint Flags:**

```
176 }  
177 val RunCoordinatedShutdownWhenDown: Boolean = cc.getBoolean("run-coordinated-shutdown-when-down")  
178 val JmxEnabled: Boolean = cc.getBoolean("jmx.enabled")  
179 val JmxMultiMbeansInSameEnabled: Boolean = cc.getBoolean("jmx.multi-mbeans-in-same-jvm")  
180 val UseDispatcher: String = cc.getString("use-dispatcher")  
181 val GossipDifferentViewProbability: Double = cc.getDouble("gossip-different-view-probability")  
182 val ReduceGossipDifferentViewProbability: Int = cc.getInt("reduce-gossip-different-view-probability")
```

main/scala/akka/cluster/ClusterSettings.scala, line 180 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:180**Taint Flags:**

```
177 val RunCoordinatedShutdownWhenDown: Boolean = cc.getBoolean("run-coordinated-shutdown-when-down")  
178 val JmxEnabled: Boolean = cc.getBoolean("jmx.enabled")  
179 val JmxMultiMbeansInSameEnabled: Boolean = cc.getBoolean("jmx.multi-mbeans-in-same-jvm")  
180 val UseDispatcher: String = cc.getString("use-dispatcher")  
181 val GossipDifferentViewProbability: Double = cc.getDouble("gossip-different-view-probability")  
182 val ReduceGossipDifferentViewProbability: Int = cc.getInt("reduce-gossip-different-view-probability")  
183 val SchedulerTickDuration: FiniteDuration = cc.getMillisDuration("scheduler.tick-duration")
```

main/scala/akka/cluster/ClusterSettings.scala, line 181 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterSettings.scala, line 181 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: main/scala/akka/cluster/ClusterSettings.scala:181

Taint Flags:

```

178 val JmxEnabled: Boolean = cc.getBoolean("jmx.enabled")
179 val JmxMultiMbeansInSameEnabled: Boolean = cc.getBoolean("jmx.multi-mbeans-in-same-jvm")
180 val UseDispatcher: String = cc.getString("use-dispatcher")
181 val GossipDifferentViewProbability: Double = cc.getDouble("gossip-different-view-probability")
182 val ReduceGossipDifferentViewProbability: Int = cc.getInt("reduce-gossip-different-view-probability")
183 val SchedulerTickDuration: FiniteDuration = cc.getMillisDuration("scheduler.tick-duration")
184 val SchedulerTicksPerWheel: Int = cc.getInt("scheduler.ticks-per-wheel")

```

main/scala/akka/cluster/ClusterSettings.scala, line 182 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc

Enclosing Method: ClusterSettings()

File: main/scala/akka/cluster/ClusterSettings.scala:182

Taint Flags:

```

179 val JmxMultiMbeansInSameEnabled: Boolean = cc.getBoolean("jmx.multi-mbeans-in-same-jvm")
180 val UseDispatcher: String = cc.getString("use-dispatcher")
181 val GossipDifferentViewProbability: Double = cc.getDouble("gossip-different-view-probability")
182 val ReduceGossipDifferentViewProbability: Int = cc.getInt("reduce-gossip-different-view-probability")
183 val SchedulerTickDuration: FiniteDuration = cc.getMillisDuration("scheduler.tick-duration")
184 val SchedulerTicksPerWheel: Int = cc.getInt("scheduler.ticks-per-wheel")
185

```

main/scala/akka/cluster/ClusterSettings.scala, line 183 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc

Enclosing Method: ClusterSettings()

File: main/scala/akka/cluster/ClusterSettings.scala:183

Taint Flags:

```

180 val UseDispatcher: String = cc.getString("use-dispatcher")

```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 183 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
181 val GossipDifferentViewProbability: Double = cc.getDouble("gossip-different-view-probability")
182 val ReduceGossipDifferentViewProbability: Int = cc.getInt("reduce-gossip-different-view-probability")
183 val SchedulerTickDuration: FiniteDuration = cc.getMillisDuration("scheduler.tick-duration")
184 val SchedulerTicksPerWheel: Int = cc.getInt("scheduler.ticks-per-wheel")
185
186 val ByPassConfigCompatCheck: Boolean = !cc.getBoolean("configuration-compatibility-check.enforce-on-join")
```

main/scala/akka/cluster/ClusterSettings.scala, line 184 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:184**Taint Flags:**

```
181 val GossipDifferentViewProbability: Double = cc.getDouble("gossip-different-view-probability")
182 val ReduceGossipDifferentViewProbability: Int = cc.getInt("reduce-gossip-different-view-probability")
183 val SchedulerTickDuration: FiniteDuration = cc.getMillisDuration("scheduler.tick-duration")
184 val SchedulerTicksPerWheel: Int = cc.getInt("scheduler.ticks-per-wheel")
185
186 val ByPassConfigCompatCheck: Boolean = !cc.getBoolean("configuration-compatibility-check.enforce-on-join")
187 val ConfigCompatCheckers: Set[String] = {
```

main/scala/akka/cluster/ClusterSettings.scala, line 186 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$ClusterSettings\$\$cc**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:186**Taint Flags:**

```
183 val SchedulerTickDuration: FiniteDuration = cc.getMillisDuration("scheduler.tick-duration")
184 val SchedulerTicksPerWheel: Int = cc.getInt("scheduler.ticks-per-wheel")
185
186 val ByPassConfigCompatCheck: Boolean = !cc.getBoolean("configuration-compatibility-check.enforce-on-join")
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterSettings.scala, line 186 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 187 val ConfigCompatCheckers: Set[String] = { 188 import akka.util.compat.JavaConverters._ 189 cc.getConfig("configuration-compatibility-check.checkers") </pre>	
main/scala/akka/cluster/ClusterSettings.scala, line 187 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:187 Taint Flags:	
<pre> 184 val SchedulerTicksPerWheel: Int = cc.getInt("scheduler.ticks-per-wheel") 185 186 val ByPassConfigCompatCheck: Boolean = !cc.getBoolean("configuration-compatibility-check.enforce-on-join") 187 val ConfigCompatCheckers: Set[String] = { 188 import akka.util.compat.JavaConverters._ 189 cc.getConfig("configuration-compatibility-check.checkers") 190 .root </pre>	
main/scala/akka/cluster/ClusterSettings.scala, line 204 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: akka\$cluster\$ClusterSettings\$\$cc Enclosing Method: ClusterSettings() File: main/scala/akka/cluster/ClusterSettings.scala:204 Taint Flags:	
<pre> 201 val SensitiveConfigPaths = { 202 import akka.util.compat.JavaConverters._ 203 204 val sensitiveKeys = 205 cc.getConfig("configuration-compatibility-check.sensitive-config-paths") 206 .root 207 .unwrapped </pre>	



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 70 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: g3**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:70**Taint Flags:**

```
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
```

test/scala/akka/cluster/JoinConfigCompatCheckClusterSpec.scala, line 17 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: clusterSettings**Enclosing Method:** JoinConfigCompatCheckClusterSpec()**File:** test/scala/akka/cluster/JoinConfigCompatCheckClusterSpec.scala:17**Taint Flags:**

```
14
15 private val extSystem = system.asInstanceOf[ExtendedActorSystem]
16 private val clusterSettings = new ClusterSettings(system.settings.config, system.name)
17 private val joinConfigCompatChecker: JoinConfigCompatChecker =
18 JoinConfigCompatChecker.load(extSystem, clusterSettings)
19
20 // Corresponding to the check of InitJoin
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 39 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/VectorClockPerfSpec.scala, line 39 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: createVectorClockOfSize**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:39**Taint Flags:**

```
36 // increase for serious measurements
37 val iterations = sys.props.get("akka.cluster.VectorClockPerfSpec.iterations").getOrElse("1000").toInt
38
39 val (vcBefore, nodes) = createVectorClockOfSize(clockSize)
40 val firstNode = nodes.head
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 40 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: nodes**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:40**Taint Flags:**

```
37 val iterations = sys.props.get("akka.cluster.VectorClockPerfSpec.iterations").getOrElse("1000").toInt
38
39 val (vcBefore, nodes) = createVectorClockOfSize(clockSize)
40 val firstNode = nodes.head
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 41 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: nodes**Enclosing Method:** VectorClockPerfSpec()

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/VectorClockPerfSpec.scala, line 41 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:41**Taint Flags:**

```
38
39 val (vcBefore, nodes) = createVectorClockOfSize(clockSize)
40 val firstNode = nodes.head
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: nodes**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:42**Taint Flags:**

```
39 val (vcBefore, nodes) = createVectorClockOfSize(clockSize)
40 val firstNode = nodes.head
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
```

test/scala/akka/cluster/ClusterSpec.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: config**Enclosing Method:** ClusterSpec()**File:** test/scala/akka/cluster/ClusterSpec.scala:51**Taint Flags:**

```
48 final case class GossipTo(address: Address)
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterSpec.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function)	Low

```

49 }
50
51 class ClusterSpec extends AkkaSpec(ClusterSpec.config) with ImplicitSender {
52
53   val selfAddress = system.asInstanceOf[ExtendedActorSystem].provider.getDefaultAddress
54

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 80 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: state
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:80
Taint Flags:

```

77 .seen(a51Up.uniqueAddress)
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
83 val g8 = Gossip(

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 64 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: g0
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:64
Taint Flags:

```

61 val emptyMembershipState = state(Gossip.empty, aUp.uniqueAddress)
62
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)
64 val state0 = state(g0, aUp.uniqueAddress)

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 64 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress) 66 val state1 = state(g1, aUp.uniqueAddress) 67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress) </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 91 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: state Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:91 Taint Flags:	
<pre> 88 val g9 = Gossip(89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp), 90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress))) 91 val state9 = state(g9, aUp.uniqueAddress) 92 val g10 = Gossip(93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp), 94 overview = GossipOverview(reachability = Reachability.empty)) </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cUp Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:53 Taint Flags:	
<pre> 50 val bRemoved = bExiting.copy(status = Removed) 51 val cJoining = TestMember(Address(protocol, "sys", "c", 2552), Joining, Set("GRP")) 52 val cUp = cJoining.copy(status = Up) 53 val cRemoved = cUp.copy(status = Removed) 54 val a51Up = TestMember(Address(protocol, "sys", "a", 2551), Up) 55 val dUp = TestMember(Address(protocol, "sys", "d", 2552), Up, Set("GRP")) 56 val eUp = TestMember(Address(protocol, "sys", "e", 2552), Up, Set("GRP"), OtherDataCenter) </pre>	

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

Package: akka.cluster

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 67 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cUp
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:67
Taint Flags:

```

64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 69 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cUp
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:69
Taint Flags:

```

66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 71 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 71 (Code Correctness: Constructor Invokes Overridable Function)	Low
Sink Details	

Sink: FunctionCall: cUp
Enclosing Method: ClusterDomainEventPublisherSpec()
File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:71
Taint Flags:

```

68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details	
Sink: FunctionCall: cUp Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:73 Taint Flags:	
<pre> 70 val state3 = state(g3, aUp.uniqueAddress) 71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress) 72 val state4 = state(g4, aUp.uniqueAddress) 73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)) 74 .seen(aUp.uniqueAddress) 75 .seen(bExiting.uniqueAddress) 76 .seen(cUp.uniqueAddress) </pre>	

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details	
Sink: FunctionCall: cUp Enclosing Method: ClusterDomainEventPublisherSpec()	



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:73**Taint Flags:**

```
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74 .seen(aUp.uniqueAddress)
75 .seen(bExiting.uniqueAddress)
76 .seen(cUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:79**Taint Flags:**

```
76 .seen(cUp.uniqueAddress)
77 .seen(a51Up.uniqueAddress)
78 val state5 = state(g5, aUp.uniqueAddress)
79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress)
80 val state6 = state(g6, aUp.uniqueAddress)
81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress)
82 val state7 = state(g7, aUp.uniqueAddress)
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 81 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cUp**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:81**Taint Flags:**

```
78 val state5 = state(g5, aUp.uniqueAddress)
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 81 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress) 80 val state6 = state(g6, aUp.uniqueAddress) 81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress) 82 val state7 = state(g7, aUp.uniqueAddress) 83 val g8 = Gossip(84 members = SortedSet(aUp, bExiting, cUp, dUp), </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 84 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cUp Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:84 Taint Flags:	
<pre> 81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress) 82 val state7 = state(g7, aUp.uniqueAddress) 83 val g8 = Gossip(84 members = SortedSet(aUp, bExiting, cUp, dUp), 85 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, dUp.uniqueAddress))) 86 .seen(aUp.uniqueAddress) 87 val state8 = state(g8, aUp.uniqueAddress) </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 89 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cUp Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:89 Taint Flags:	
<pre> 86 .seen(aUp.uniqueAddress) 87 val state8 = state(g8, aUp.uniqueAddress) 88 val g9 = Gossip(89 members = SortedSet(aUp, bExiting, cUp, dUp, eUp), </pre>	



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 89 (Code Correctness: Constructor Invokes Overridable Function)	Low
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress))) 91 val state9 = state(g9, aUp.uniqueAddress) 92 val g10 = Gossip(
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 93 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cUp Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:93 Taint Flags:	
90 overview = GossipOverview(reachability = Reachability.empty.unreachable(aUp.uniqueAddress, eUp.uniqueAddress))) 91 val state9 = state(g9, aUp.uniqueAddress) 92 val g10 = Gossip(93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp), 94 overview = GossipOverview(reachability = Reachability.empty)) 95 val state10 = state(g10, aUp.uniqueAddress) 96	
main/scala/akka/cluster/Cluster.scala, line 172 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: akka\$cluster\$Cluster\$\$log Enclosing Method: Cluster() File: main/scala/akka/cluster/Cluster.scala:172 Taint Flags:	
169 .createInstanceFor[Scheduler](170 system.settings.SchedulerClass, 171 immutable 172 .Seq(classOf[Config] -> cfg, classOf[LoggingAdapter] -> log, classOf[ThreadFactory] -> threadFactory)) 173 .get 174 } else { 175 // delegate to system.scheduler, but don't close over system	



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/Cluster.scala, line 225 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: akka\$cluster\$Cluster\$\$log**Enclosing Method:** Cluster()**File:** main/scala/akka/cluster/Cluster.scala:225**Taint Flags:**

```
222
223 if (JmxEnabled)
224   clusterJmx = {
225     val jmx = new ClusterJmx(this, log)
226     jmx.createMBean()
227     Some(jmx)
228   }
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 72 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: g4**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:72**Taint Flags:**

```
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
74   .seen(aUp.uniqueAddress)
75   .seen(bExiting.uniqueAddress)
```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 128 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterHeartbeat.scala, line 128 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: failureDetector**Enclosing Method:** ClusterHeartbeatSender()**File:** main/scala/akka/cluster/ClusterHeartbeat.scala:128**Taint Flags:**

125

126 val failureDetector = cluster.failureDetector

127

128 var state: ClusterHeartbeatSenderState = ClusterHeartbeatSenderState(

129 ring = HeartbeatNodeRing(selfUniqueAddress, Set(selfUniqueAddress), Set.empty, MonitoredByNrOfMembers),

130 oldReceiversNowUnreachable = Set.empty[UniqueAddress],

131 failureDetector)

test/scala/akka/cluster/JoinConfigCompatCheckerSpec.scala, line 26 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: baseConfig**Enclosing Method:** JoinConfigCompatCheckerSpec()**File:** test/scala/akka/cluster/JoinConfigCompatCheckerSpec.scala:26**Taint Flags:**

23 akka.remote.artery.advanced.aeron.idle-cpu-level = 3

24 """)

25

26 val configWithChecker: Config =

27 ConfigFactory.parseString("""

28 akka.cluster {

29 config-compat-test = "test"

test/scala/akka/cluster/HeartbeatNodeRingSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: ff**Enclosing Method:** HeartbeatNodeRingSpec()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/HeartbeatNodeRingSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/HeartbeatNodeRingSpec.scala:21
Taint Flags:

```

18 val ee = UniqueAddress(Address("akka", "sys", "ee", 2552), 5L)
19 val ff = UniqueAddress(Address("akka", "sys", "ff", 2552), 6L)
20
21 val nodes = Set(aa, bb, cc, dd, ee, ff)
22
23 "A HashedNodeRing" must {
24
```

test/scala/akka/cluster/ClusterDeathWatchNotificationSpec.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: system3
Enclosing Method: ClusterDeathWatchNotificationSpec()
File: test/scala/akka/cluster/ClusterDeathWatchNotificationSpec.scala:53
Taint Flags:

```

50 private def system1: ActorSystem = system
51 private val system2 = newRemoteSystem(name = Some(system.name))
52 private val system3 = newRemoteSystem(name = Some(system.name))
53 private val systems = Vector(system1, system2, system3)
54
55 private val messages = (1 to 100).map(_._toString).toVector
56
```

main/scala/akka/cluster/ClusterSettings.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: LogInfoVerbose
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:42
Taint Flags:

```

39 private val cc = config.getConfig("akka.cluster")
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
40
41 val LogInfoVerbose: Boolean = cc.getBoolean("log-info-verbose")
42 val LogInfo: Boolean = LogInfoVerbose || cc.getBoolean("log-info")
43 val FailureDetectorConfig: Config = cc.getConfig("failure-detector")
44 val FailureDetectorImplementationClass: String = FailureDetectorConfig.getString("implementation-class")
45 val HeartbeatInterval: FiniteDuration = {
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 70 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: state**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:70**Taint Flags:**

```
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
69 val g3 = g2.seen(bExiting.uniqueAddress).seen(cUp.uniqueAddress)
70 val state3 = state(g3, aUp.uniqueAddress)
71 val g4 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
72 val state4 = state(g4, aUp.uniqueAddress)
73 val g5 = Gossip(members = SortedSet(a51Up, aUp, bExiting, cUp))
```

test/scala/akka/cluster/HeartbeatNodeRingSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aa**Enclosing Method:** HeartbeatNodeRingSpec()**File:** test/scala/akka/cluster/HeartbeatNodeRingSpec.scala:21**Taint Flags:**

```
18 val ee = UniqueAddress(Address("akka", "sys", "ee", 2552), 5L)
19 val ff = UniqueAddress(Address("akka", "sys", "ff", 2552), 6L)
20
21 val nodes = Set(aa, bb, cc, dd, ee, ff)
```



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

Package: akka.cluster

test/scala/akka/cluster/HeartbeatNodeRingSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```

22
23 "A HashedNodeRing" must {
24

```

test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: v1Config
Enclosing Method: JoinConfigCompatCheckerRollingUpdateSpec()
File: test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala:42
Taint Flags:

```

39 }
40 """)
41
42 val v2Config: Config = v2.withFallback(v1Config)
43
44 val v2ConfigIncompatible: Config = v2.withFallback(baseConfig)
45

```

test/scala/akka/cluster/ShutdownAfterJoinSeedNodesSpec.scala, line 31 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: config
Enclosing Method: ShutdownAfterJoinSeedNodesSpec()
File: test/scala/akka/cluster/ShutdownAfterJoinSeedNodesSpec.scala:31
Taint Flags:

```

28 """)
29 }
30
31 class ShutdownAfterJoinSeedNodesSpec extends AkkaSpec(ShutdownAfterJoinSeedNodesSpec.config) {
32
33 val seed1 = ActorSystem(system.name, system.settings.config)
34 val seed2 = ActorSystem(system.name, system.settings.config)

```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 19 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:19**Taint Flags:**

```
16 import MemberStatus._
17
18 val aRoles = Set("AA", "AB")
19 val aJoining = TestMember(Address("akka", "sys", "a", 2552), Joining, aRoles)
20 val aUp = TestMember(Address("akka", "sys", "a", 2552), Up, aRoles)
21 val aRemoved = TestMember(Address("akka", "sys", "a", 2552), Removed, aRoles)
22 val bRoles = Set("AB", "BB")
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 20 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: aRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:20**Taint Flags:**

```
17
18 val aRoles = Set("AA", "AB")
19 val aJoining = TestMember(Address("akka", "sys", "a", 2552), Joining, aRoles)
20 val aUp = TestMember(Address("akka", "sys", "a", 2552), Up, aRoles)
21 val aRemoved = TestMember(Address("akka", "sys", "a", 2552), Removed, aRoles)
22 val bRoles = Set("AB", "BB")
23 val bUp = TestMember(Address("akka", "sys", "b", 2552), Up, bRoles)
```

test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventSpec.scala, line 21 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: aRoles**Enclosing Method:** ClusterDomainEventSpec()**File:** test/scala/akka/cluster/ClusterDomainEventSpec.scala:21**Taint Flags:**

```
18 val aRoles = Set("AA", "AB")
19 val aJoining = TestMember(Address("akka", "sys", "a", 2552), Joining, aRoles)
20 val aUp = TestMember(Address("akka", "sys", "a", 2552), Up, aRoles)
21 val aRemoved = TestMember(Address("akka", "sys", "a", 2552), Removed, aRoles)
22 val bRoles = Set("AB", "BB")
23 val bUp = TestMember(Address("akka", "sys", "b", 2552), Up, bRoles)
24 val bDown = TestMember(Address("akka", "sys", "b", 2552), Down, bRoles)
```

main/scala/akka/cluster/ClusterEvent.scala, line 674 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** ClusterDomainEventPublisher()**File:** main/scala/akka/cluster/ClusterEvent.scala:674**Taint Flags:**

```
671 import InternalClusterAction._
672
673 val cluster = Cluster(context.system)
674 val selfUniqueAddress = cluster.selfUniqueAddress
675 val emptyMembershipState = MembershipState(
676   Gossip.empty,
677   cluster.selfUniqueAddress,
```

main/scala/akka/cluster/ClusterEvent.scala, line 675 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cluster**Enclosing Method:** ClusterDomainEventPublisher()

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterEvent.scala, line 675 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** main/scala/akka/cluster/ClusterEvent.scala:675**Taint Flags:**

```
672
673 val cluster = Cluster(context.system)
674 val selfUniqueAddress = cluster.selfUniqueAddress
675 val emptyMembershipState = MembershipState(
676   Gossip.empty,
677   cluster.selfUniqueAddress,
678   cluster.settings.SelfDataCenter,
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: vcBefore**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:43**Taint Flags:**

```
40 val firstNode = nodes.head
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: vcBefore**Enclosing Method:** VectorClockPerfSpec()**File:** test/scala/akka/cluster/VectorClockPerfSpec.scala:46**Taint Flags:**

```
43 val vcBaseLast = vcBefore :+ lastNode
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/VectorClockPerfSpec.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function)****Low**

```
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
48 val vcConcurrentMiddle = vcBaseMiddle :+ middleNode
49
```

test/scala/akka/cluster/JoinConfigCompatCheckClusterSpec.scala, line 17 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: extSystem**Enclosing Method:** JoinConfigCompatCheckClusterSpec()**File:** test/scala/akka/cluster/JoinConfigCompatCheckClusterSpec.scala:17**Taint Flags:**

```
14
15 private val extSystem = system.asInstanceOf[ExtendedActorSystem]
16 private val clusterSettings = new ClusterSettings(system.settings.config, system.name)
17 private val joinConfigCompatChecker: JoinConfigCompatChecker =
18 JoinConfigCompatChecker.load(extSystem, clusterSettings)
19
20 // Corresponding to the check of InitJoin
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 78 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: state**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:78**Taint Flags:**

```
75 .seen(bExiting.uniqueAddress)
76 .seen(cUp.uniqueAddress)
77 .seen(a51Up.uniqueAddress)
78 val state5 = state(g5, aUp.uniqueAddress)
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 78 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 79 val g6 = Gossip(members = SortedSet(aLeaving, bExiting, cUp)).seen(aUp.uniqueAddress) 80 val state6 = state(g6, aUp.uniqueAddress) 81 val g7 = Gossip(members = SortedSet(aExiting, bExiting, cUp)).seen(aUp.uniqueAddress) </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 52 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cJoining Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:52 Taint Flags:	
<pre> 49 val bExiting = TestMember(Address(protocol, "sys", "b", 2552), Exiting) 50 val bRemoved = bExiting.copy(status = Removed) 51 val cJoining = TestMember(Address(protocol, "sys", "c", 2552), Joining, Set("GRP")) 52 val cUp = cJoining.copy(status = Up) 53 val cRemoved = cUp.copy(status = Removed) 54 val a51Up = TestMember(Address(protocol, "sys", "a", 2551), Up) 55 val dUp = TestMember(Address(protocol, "sys", "d", 2552), Up, Set("GRP")) </pre>	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 65 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cJoining Enclosing Method: ClusterDomainEventPublisherSpec() File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:65 Taint Flags:	
<pre> 62 63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress) 64 val state0 = state(g0, aUp.uniqueAddress) 65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress) 66 val state1 = state(g1, aUp.uniqueAddress) 67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress) 68 val state2 = state(g2, aUp.uniqueAddress) </pre>	

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 65 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cJoining**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:65**Taint Flags:**

```
62
63 val g0 = Gossip(members = SortedSet(aUp)).seen(aUp.uniqueAddress)
64 val state0 = state(g0, aUp.uniqueAddress)
65 val g1 = Gossip(members = SortedSet(aUp, cJoining)).seen(aUp.uniqueAddress).seen(cJoining.uniqueAddress)
66 val state1 = state(g1, aUp.uniqueAddress)
67 val g2 = Gossip(members = SortedSet(aUp, bExiting, cUp)).seen(aUp.uniqueAddress)
68 val state2 = state(g2, aUp.uniqueAddress)
```

main/scala/akka/cluster/ClusterSettings.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: FailureDetectorConfig**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:44**Taint Flags:**

```
41 val LogInfoVerbose: Boolean = cc.getBoolean("log-info-verbose")
42 val LogInfo: Boolean = LogInfoVerbose || cc.getBoolean("log-info")
43 val FailureDetectorConfig: Config = cc.getConfig("failure-detector")
44 val FailureDetectorImplementationClass: String = FailureDetectorConfig.getString("implementation-class")
45 val HeartbeatInterval: FiniteDuration = {
46 FailureDetectorConfig.getMillisDuration("heartbeat-interval")
47 }.requiring(_ > Duration.Zero, "failure-detector.heartbeat-interval must be > 0")
```

main/scala/akka/cluster/ClusterSettings.scala, line 45 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 45 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details**

Sink: FunctionCall: FailureDetectorConfig
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:45
Taint Flags:

```
42 val LogInfo: Boolean = LogInfoVerbose || cc.getBoolean("log-info")
43 val FailureDetectorConfig: Config = cc.getConfig("failure-detector")
44 val FailureDetectorImplementationClass: String = FailureDetectorConfig.getString("implementation-class")
45 val HeartbeatInterval: FiniteDuration = {
46 FailureDetectorConfig.getMillisDuration("heartbeat-interval")
47 }.requiring(_ > Duration.Zero, "failure-detector.heartbeat-interval must be > 0")
48 val HeartbeatExpectedResponseAfter: FiniteDuration = {
```

main/scala/akka/cluster/ClusterSettings.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: FailureDetectorConfig
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:48
Taint Flags:

```
45 val HeartbeatInterval: FiniteDuration = {
46 FailureDetectorConfig.getMillisDuration("heartbeat-interval")
47 }.requiring(_ > Duration.Zero, "failure-detector.heartbeat-interval must be > 0")
48 val HeartbeatExpectedResponseAfter: FiniteDuration = {
49 FailureDetectorConfig.getMillisDuration("expected-response-after")
50 }.requiring(_ > Duration.Zero, "failure-detector.expected-response-after > 0")
51 val MonitoredByNrOfMembers: Int = {
```

main/scala/akka/cluster/ClusterSettings.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: FailureDetectorConfig
Enclosing Method: ClusterSettings()



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function)****Low****File:** main/scala/akka/cluster/ClusterSettings.scala:51**Taint Flags:**

```
48 val HeartbeatExpectedResponseAfter: FiniteDuration = {  
49 FailureDetectorConfig.getMillisDuration("expected-response-after")  
50 }.requiring(_ > Duration.Zero, "failure-detector.expected-response-after > 0")  
51 val MonitoredByNrOfMembers: Int = {  
52 FailureDetectorConfig.getInt("monitored-by-nr-of-members")  
53 }.requiring(_ > 0, "failure-detector.monitored-by-nr-of-members must be > 0")  
54
```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 95 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: state**Enclosing Method:** ClusterDomainEventPublisherSpec()**File:** test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:95**Taint Flags:**

```
92 val g10 = Gossip(  
93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp),  
94 overview = GossipOverview(reachability = Reachability.empty))  
95 val state10 = state(g10, aUp.uniqueAddress)  
96  
97 // created in beforeEach  
98 var memberSubscriber: TestProbe = _
```

main/scala/akka/cluster/Gossip.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: assertInvariants**Enclosing Method:** Gossip()**File:** main/scala/akka/cluster/Gossip.scala:73**Taint Flags:**

```
70 version: VectorClock = VectorClock(), // vector clock version
```



Code Correctness: Constructor Invokes Overridable Function		Low
Package: akka.cluster		
main/scala/akka/cluster/Gossip.scala, line 73 (Code Correctness: Constructor Invokes Overridable Function)		Low
<pre> 71 tombstones: Map[UniqueAddress, Gossip.Timestamp] = Map.empty) { 72 73 if (Cluster.isAssertInvariantsEnabled) assertInvariants() 74 75 private def assertInvariants(): Unit = { 76 </pre>		
test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function)		Low
Issue Details		
Kingdom: Code Quality Scan Engine: SCA (Structural)		
Sink Details		
Sink: FunctionCall: baseConfig Enclosing Method: JoinConfigCompatCheckerRollingUpdateSpec() File: test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala:33 Taint Flags:		
<pre> 30 } 31 """).withFallback(JoinConfigCompatCheckerSpec.baseConfig) 32 33 val v1Config: Config = baseConfig.withFallback(JoinConfigCompatCheckerSpec.configWithChecker) 34 35 private val v2 = ConfigFactory.parseString(""" 36 akka.cluster.new-configuration = "v2" </pre>		
test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)		Low
Issue Details		
Kingdom: Code Quality Scan Engine: SCA (Structural)		
Sink Details		
Sink: FunctionCall: baseConfig Enclosing Method: JoinConfigCompatCheckerRollingUpdateSpec() File: test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala:44 Taint Flags:		
<pre> 41 42 val v2Config: Config = v2.withFallback(v1Config) 43 44 val v2ConfigIncompatible: Config = v2.withFallback(baseConfig) </pre>		



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

Package: akka.cluster

test/scala/akka/cluster/JoinConfigCompatCheckerRollingUpdateSpec.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```
45
46 }
47
```

test/scala/akka/cluster/GossipSpec.scala, line 29 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: e1
Enclosing Method: GossipSpec()
File: test/scala/akka/cluster/GossipSpec.scala:29
Taint Flags:

```
26 val c3 = TestMember(c1.address, Exiting)
27 val d1 = TestMember(Address("akka", "sys", "d", 2552), Leaving)
28 val e1 = TestMember(Address("akka", "sys", "e", 2552), Joining)
29 val e2 = TestMember(e1.address, Up)
30 val e3 = TestMember(e1.address, Down)
31 val f1 = TestMember(Address("akka", "sys", "f", 2552), Joining)
32
```

test/scala/akka/cluster/GossipSpec.scala, line 30 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: e1
Enclosing Method: GossipSpec()
File: test/scala/akka/cluster/GossipSpec.scala:30
Taint Flags:

```
27 val d1 = TestMember(Address("akka", "sys", "d", 2552), Leaving)
28 val e1 = TestMember(Address("akka", "sys", "e", 2552), Joining)
29 val e2 = TestMember(e1.address, Up)
30 val e3 = TestMember(e1.address, Down)
31 val f1 = TestMember(Address("akka", "sys", "f", 2552), Joining)
32
33 val dc1a1 = TestMember(Address("akka", "sys", "a", 2552), Up, Set.empty, dataCenter = "dc1")
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster**test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: addUnreachable**Enclosing Method:** ReachabilityPerfSpec()**File:** test/scala/akka/cluster/ReachabilityPerfSpec.scala:43**Taint Flags:**

```
40
41 val reachability1 = createReachabilityOfSize(Reachability.empty, nodesSize)
42 val reachability2 = createReachabilityOfSize(reachability1, nodesSize)
43 val reachability3 = addUnreachable(reachability1, nodesSize / 2)
44 val allowed = reachability1.versions.keySet
45
46 private def checkThunkFor(
```

test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 41 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: nodesSize**Enclosing Method:** ReachabilityPerfSpec()**File:** test/scala/akka/cluster/ReachabilityPerfSpec.scala:41**Taint Flags:**

```
38 }
39 }
40
41 val reachability1 = createReachabilityOfSize(Reachability.empty, nodesSize)
42 val reachability2 = createReachabilityOfSize(reachability1, nodesSize)
43 val reachability3 = addUnreachable(reachability1, nodesSize / 2)
44 val allowed = reachability1.versions.keySet
```

test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)	Low

Sink Details

Sink: FunctionCall: nodesSize
Enclosing Method: ReachabilityPerfSpec()
File: test/scala/akka/cluster/ReachabilityPerfSpec.scala:42
Taint Flags:

```

39 }
40
41 val reachability1 = createReachabilityOfSize(Reachability.empty, nodesSize)
42 val reachability2 = createReachabilityOfSize(reachability1, nodesSize)
43 val reachability3 = addUnreachable(reachability1, nodesSize / 2)
44 val allowed = reachability1.versions.keySet
45

```

test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodesSize
Enclosing Method: ReachabilityPerfSpec()
File: test/scala/akka/cluster/ReachabilityPerfSpec.scala:43
Taint Flags:

```

40
41 val reachability1 = createReachabilityOfSize(Reachability.empty, nodesSize)
42 val reachability2 = createReachabilityOfSize(reachability1, nodesSize)
43 val reachability3 = addUnreachable(reachability1, nodesSize / 2)
44 val allowed = reachability1.versions.keySet
45
46 private def checkThunkFor(

```

test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 90 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodesSize
Enclosing Method: ReachabilityPerfSpec()



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ReachabilityPerfSpec.scala, line 90 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/ReachabilityPerfSpec.scala:90

Taint Flags:

```

87 }
88 }
89
90 s"Reachability of size $nodesSize" must {
91
92 s"do a warm up run, $iterations times" in {
93 checkThunkFor(reachability1, reachability2, merge(0), iterations)

```

test/scala/akka/cluster/ClusterDeployerSpec.scala, line 50 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: deployerConf

Enclosing Method: ClusterDeployerSpec()

File: test/scala/akka/cluster/ClusterDeployerSpec.scala:50

Taint Flags:

```

47
48 }
49
50 class ClusterDeployerSpec extends AkkaSpec(ClusterDeployerSpec.deployerConf) {
51
52 "A RemoteDeployer" must {
53

```

test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 95 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: g10

Enclosing Method: ClusterDomainEventPublisherSpec()

File: test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala:95

Taint Flags:

```

92 val g10 = Gossip(

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster	
test/scala/akka/cluster/ClusterDomainEventPublisherSpec.scala, line 95 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 93 members = SortedSet(aUp, bExiting, cUp, dUp, eUp), 94 overview = GossipOverview(reachability = Reachability.empty)) 95 val state10 = state(g10, aUp.uniqueAddress) 96 97 // created in beforeEach 98 var memberSubscriber: TestProbe = _ </pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 209 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: coordShutdown Enclosing Method: ClusterDaemon() File: main/scala/akka/cluster/ClusterDaemon.scala:209 Taint Flags:	
<pre> 206 207 val clusterShutdown = Promise[Done]() 208 val coordShutdown = CoordinatedShutdown(context.system) 209 coordShutdown.addTask(CoordinatedShutdown.PhaseClusterLeave, "leave") { 210 val sys = context.system 211 } => 212 if (Cluster(sys).isTerminated Cluster(sys).selfMember.status == Down) </pre>	
main/scala/akka/cluster/ClusterDaemon.scala, line 219 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: coordShutdown Enclosing Method: ClusterDaemon() File: main/scala/akka/cluster/ClusterDaemon.scala:219 Taint Flags:	
<pre> 216 self.ask(CoordinatedShutdownLeave.LeaveReq).mapTo[Done] 217 } 218 } 219 coordShutdown.addTask(CoordinatedShutdown.PhaseClusterShutdown, "wait-shutdown") { () => </pre>	



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

Package: akka.cluster

main/scala/akka/cluster/ClusterDaemon.scala, line 219 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

```
220 clusterShutdown.future
221 }
222
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: vcBaseLast
Enclosing Method: VectorClockPerfSpec()
File: test/scala/akka/cluster/VectorClockPerfSpec.scala:44
Taint Flags:

```
41 val lastNode = nodes.last
42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
```

test/scala/akka/cluster/VectorClockPerfSpec.scala, line 45 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: vcBaseLast
Enclosing Method: VectorClockPerfSpec()
File: test/scala/akka/cluster/VectorClockPerfSpec.scala:45
Taint Flags:

```
42 val middleNode = nodes.drop(clockSize / 2).head
43 val vcBaseLast = vcBefore :+ lastNode
44 val vcAfterLast = vcBaseLast :+ firstNode
45 val vcConcurrentLast = vcBaseLast :+ lastNode
46 val vcBaseMiddle = vcBefore :+ middleNode
47 val vcAfterMiddle = vcBaseMiddle :+ firstNode
48 val vcConcurrentMiddle = vcBaseMiddle :+ middleNode
```



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster.protobuf**main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 388 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: reachabilityStatusToInt**Enclosing Method:** ClusterMessageSerializer()**File:** main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:388**Taint Flags:**

```
385 Reachability.Unreachable -> cm.ReachabilityStatus.Unreachable_VALUE,  
386 Reachability.Terminated -> cm.ReachabilityStatus.Terminated_VALUE)  
387  
388 private val reachabilityStatusFromInt = reachabilityStatusToInt.map { case (a, b) => (b, a) }  
389  
390 private def mapWithErrorMessage[T](map: Map[T, Int], value: T, unknown: String): Int = map.get(value) match {  
391 case Some(x) => x
```

main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala, line 381 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: memberStatusToInt**Enclosing Method:** ClusterMessageSerializer()**File:** main/scala/akka/cluster/protobuf/ClusterMessageSerializer.scala:381**Taint Flags:**

```
378 MemberStatus.PreparingForShutdown -> cm.MemberStatus.PreparingForShutdown_VALUE,  
379 MemberStatus.ReadyForShutdown -> cm.MemberStatus.ReadyForShutdown_VALUE)  
380  
381 private val memberStatusFromInt = memberStatusToInt.map { case (a, b) => (b, a) }  
382  
383 private val reachabilityStatusToInt = scala.collection.immutable.HashMap[Reachability.ReachabilityStatus, Int](  
384 Reachability.Reachable -> cm.ReachabilityStatus.Reachable_VALUE,
```

Package: akka.cluster.sbr**test/scala/akka/cluster/sbr/TestAddresses.scala, line 26 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality

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

Package: akka.cluster.sbr

test/scala/akka/cluster/sbr/TestAddresses.scala, line 26 (Code Correctness: Constructor Invokes Overridable Function)**Low**

Scan Engine: SCA (Structural)

Sink Details**Sink:** FunctionCall: addressA**Enclosing Method:** TestAddresses()**File:** test/scala/akka/cluster/sbr/TestAddresses.scala:26**Taint Flags:**

```
23 private def defaultDcRole = dcRole(defaultDataCenter)
24
25 val addressA = Address("akka.tcp", "sys", "a", 2552)
26 val memberA = new Member(UniqueAddress(addressA, 0L), 5, Up, Set("role3", defaultDcRole), Version.Zero)
27 val memberB =
28 new Member(
29 UniqueAddress(addressA.copy(host = Some("b")), 0L),
```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 28 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: addressA**Enclosing Method:** TestAddresses()**File:** test/scala/akka/cluster/sbr/TestAddresses.scala:28**Taint Flags:**

```
25 val addressA = Address("akka.tcp", "sys", "a", 2552)
26 val memberA = new Member(UniqueAddress(addressA, 0L), 5, Up, Set("role3", defaultDcRole), Version.Zero)
27 val memberB =
28 new Member(
29 UniqueAddress(addressA.copy(host = Some("b")), 0L),
30 4,
31 Up,
```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 29 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details**

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/TestAddresses.scala, line 29 (Code Correctness: Constructor Invokes Overridable Function)	Low

Sink: FunctionCall: addressA
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:29
Taint Flags:

```

26 val memberA = new Member(UniqueAddress(addressA, 0L), 5, Up, Set("role3", defaultDcRole), Version.Zero)
27 val memberB =
28 new Member(
29 UniqueAddress(addressA.copy(host = Some("b")), 0L),
30 4,
31 Up,
32 Set("role1", "role3", defaultDcRole),

```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: addressA
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:35
Taint Flags:

```

32 Set("role1", "role3", defaultDcRole),
33 Version.Zero)
34 val memberC =
35 new Member(UniqueAddress(addressA.copy(host = Some("c")), 0L), 3, Up, Set("role2", defaultDcRole), Version.Zero)
36 val memberD =
37 new Member(
38 UniqueAddress(addressA.copy(host = Some("d")), 0L),

```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 37 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: addressA
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:37
Taint Flags:



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

Package: akka.cluster.sbr

test/scala/akka/cluster/sbr/TestAddresses.scala, line 37 (Code Correctness: Constructor Invokes Overridable Function)**Low**

```
34 val memberC =  
35 new Member(UniqueAddress(addressA.copy(host = Some("c")), 0L), 3, Up, Set("role2", defaultDcRole), Version.Zero)  
36 val memberD =  
37 new Member(  
38 UniqueAddress(addressA.copy(host = Some("d")), 0L),  
39 2,  
40 Up,
```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 38 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: addressA
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:38
Taint Flags:

```
35 new Member(UniqueAddress(addressA.copy(host = Some("c")), 0L), 3, Up, Set("role2", defaultDcRole), Version.Zero)  
36 val memberD =  
37 new Member(  
38 UniqueAddress(addressA.copy(host = Some("d")), 0L),  
39 2,  
40 Up,  
41 Set("role1", "role2", "role3", defaultDcRole),
```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details**

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: addressA
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:44
Taint Flags:

```
41 Set("role1", "role2", "role3", defaultDcRole),  
42 Version.Zero)  
43 val memberE =
```



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

Package: akka.cluster.sbr

test/scala/akka/cluster/sbr/TestAddresses.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```

44 new Member(UniqueAddress(addressA.copy(host = Some("e")), 0L), 1, Up, Set(defaultDcRole), Version.Zero)
45 val memberF =
46 new Member(UniqueAddress(addressA.copy(host = Some("f")), 0L), 5, Up, Set(defaultDcRole), Version.Zero)
47 val memberG =

```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: addressA
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:46
Taint Flags:

```

43 val memberE =
44 new Member(UniqueAddress(addressA.copy(host = Some("e")), 0L), 1, Up, Set(defaultDcRole), Version.Zero)
45 val memberF =
46 new Member(UniqueAddress(addressA.copy(host = Some("f")), 0L), 5, Up, Set(defaultDcRole), Version.Zero)
47 val memberG =
48 new Member(UniqueAddress(addressA.copy(host = Some("g")), 0L), 6, Up, Set(defaultDcRole), Version.Zero)
49

```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: addressA
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:48
Taint Flags:

```

45 val memberF =
46 new Member(UniqueAddress(addressA.copy(host = Some("f")), 0L), 5, Up, Set(defaultDcRole), Version.Zero)
47 val memberG =
48 new Member(UniqueAddress(addressA.copy(host = Some("g")), 0L), 6, Up, Set(defaultDcRole), Version.Zero)
49
50 val memberAWeaklyUp = new Member(memberA.uniqueAddress, Int.MaxValue, WeaklyUp, memberA.roles, Version.Zero)

```



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

Package: akka.cluster.sbr

test/scala/akka/cluster/sbr/TestAddresses.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```
51 val memberBWeaklyUp = new Member(memberB.uniqueAddress, Int.MaxValue, WeaklyUp, memberB.roles, Version.Zero)
```

main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 60 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: DowningStableAfter

Enclosing Method: SplitBrainResolverSettings()

File: main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:60

Taint Flags:

```
57 Helpers.toRootLowerCase(cc.getString("down-all-when-unstable")) match {
58 case "on" =>
59 // based on stable-after
60 4.seconds.max(DowningStableAfter * 3 / 4)
61 case "off" =>
62 // disabled
63 Duration.Zero
```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: memberB

Enclosing Method: TestAddresses()

File: test/scala/akka/cluster/sbr/TestAddresses.scala:51

Taint Flags:

```
48 new Member(UniqueAddress(addressA.copy(host = Some("g")), 0L), 6, Up, Set(defaultDcRole), Version.Zero)
49
50 val memberAWeaklyUp = new Member(memberA.uniqueAddress, Int.MaxValue, WeaklyUp, memberA.roles, Version.Zero)
51 val memberBWeaklyUp = new Member(memberB.uniqueAddress, Int.MaxValue, WeaklyUp, memberB.roles, Version.Zero)
52
53 def dcMember(dc: ClusterSettings.DataCenter, m: Member): Member =
54 new Member(
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/TestAddresses.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function)	Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: memberB
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:51
Taint Flags:

```

48 new Member(UniqueAddress(addressA.copy(host = Some("g")), 0L), 6, Up, Set(defaultDcRole), Version.Zero)
49
50 val memberAWeaklyUp = new Member(memberA.uniqueAddress, Int.MaxValue, WeaklyUp, memberA.roles, Version.Zero)
51 val memberBWeaklyUp = new Member(memberB.uniqueAddress, Int.MaxValue, WeaklyUp, memberB.roles, Version.Zero)
52
53 def dcMember(dc: ClusterSettings.DataCenter, m: Member): Member =
54 new Member(

```

main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 49 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: allStrategyNames
Enclosing Method: SplitBrainResolverSettings()
File: main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:49
Taint Flags:

```

46
47 val DowningStrategy: String =
48 cc.getString("active-strategy").toLowerCase(Locale.ROOT) match {
49 case strategyName if allStrategyNames(strategyName) => strategyName
50 case unknown =>
51 throw new ConfigurationException(
52 s"Unknown downing strategy [$unknown]. Select one of [{allStrategyNames.mkString(", ")}]")

```

main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 51 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: allStrategyNames**Enclosing Method:** SplitBrainResolverSettings()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:51**Taint Flags:**

```
48 cc.getString("active-strategy").toLowerCase(Locale.ROOT) match {
49 case strategyName if allStrategyNames(strategyName) => strategyName
50 case unknown =>
51 throw new ConfigurationException(
52 s"Unknown downing strategy [$unknown]. Select one of [{allStrategyNames.mkString(", ")}]")
53 }
54
```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 26 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: defaultDcRole**Enclosing Method:** TestAddresses()**File:** test/scala/akka/cluster/sbr/TestAddresses.scala:26**Taint Flags:**

```
23 private def defaultDcRole = dcRole(defaultDataCenter)
24
25 val addressA = Address("akka.tcp", "sys", "a", 2552)
26 val memberA = new Member(UniqueAddress(addressA, 0L), 5, Up, Set("role3", defaultDcRole), Version.Zero)
27 val memberB =
28 new Member(
29 UniqueAddress(addressA.copy(host = Some("b")), 0L),
```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 32 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: defaultDcRole**Enclosing Method:** TestAddresses()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/TestAddresses.scala, line 32 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: test/scala/akka/cluster/sbr/TestAddresses.scala:32

Taint Flags:

```

29 UniqueAddress(addressA.copy(host = Some("b")), 0L),
30 4,
31 Up,
32 Set("role1", "role3", defaultDcRole),
33 Version.Zero)
34 val memberC =
35 new Member(UniqueAddress(addressA.copy(host = Some("c")), 0L), 3, Up, Set("role2", defaultDcRole), Version.Zero)

```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: defaultDcRole

Enclosing Method: TestAddresses()

File: test/scala/akka/cluster/sbr/TestAddresses.scala:35

Taint Flags:

```

32 Set("role1", "role3", defaultDcRole),
33 Version.Zero)
34 val memberC =
35 new Member(UniqueAddress(addressA.copy(host = Some("c")), 0L), 3, Up, Set("role2", defaultDcRole), Version.Zero)
36 val memberD =
37 new Member(
38 UniqueAddress(addressA.copy(host = Some("d")), 0L),

```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 41 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: defaultDcRole

Enclosing Method: TestAddresses()

File: test/scala/akka/cluster/sbr/TestAddresses.scala:41

Taint Flags:

```

38 UniqueAddress(addressA.copy(host = Some("d")), 0L),

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/TestAddresses.scala, line 41 (Code Correctness: Constructor Invokes Overridable Function)	Low

```

39 2,
40 Up,
41 Set("role1", "role2", "role3", defaultDcRole),
42 Version.Zero)
43 val memberE =
44 new Member(UniqueAddress(addressA.copy(host = Some("e")), 0L), 1, Up, Set(defaultDcRole), Version.Zero)

```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 44 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: defaultDcRole
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:44
Taint Flags:

```

41 Set("role1", "role2", "role3", defaultDcRole),
42 Version.Zero)
43 val memberE =
44 new Member(UniqueAddress(addressA.copy(host = Some("e")), 0L), 1, Up, Set(defaultDcRole), Version.Zero)
45 val memberF =
46 new Member(UniqueAddress(addressA.copy(host = Some("f")), 0L), 5, Up, Set(defaultDcRole), Version.Zero)
47 val memberG =

```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: defaultDcRole
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:46
Taint Flags:

```

43 val memberE =
44 new Member(UniqueAddress(addressA.copy(host = Some("e")), 0L), 1, Up, Set(defaultDcRole), Version.Zero)
45 val memberF =
46 new Member(UniqueAddress(addressA.copy(host = Some("f")), 0L), 5, Up, Set(defaultDcRole), Version.Zero)

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/TestAddresses.scala, line 46 (Code Correctness: Constructor Invokes Overridable Function)	Low
<pre> 47 val memberG = 48 new Member(UniqueAddress(addressA.copy(host = Some("g")), 0L), 6, Up, Set(defaultDcRole), Version.Zero) 49 </pre>	
test/scala/akka/cluster/sbr/TestAddresses.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: defaultDcRole Enclosing Method: TestAddresses() File: test/scala/akka/cluster/sbr/TestAddresses.scala:48 Taint Flags:	
<pre> 45 val memberF = 46 new Member(UniqueAddress(addressA.copy(host = Some("f")), 0L), 5, Up, Set(defaultDcRole), Version.Zero) 47 val memberG = 48 new Member(UniqueAddress(addressA.copy(host = Some("g")), 0L), 6, Up, Set(defaultDcRole), Version.Zero) 49 50 val memberAWeaklyUp = new Member(memberA.uniqueAddress, Int.MaxValue, WeaklyUp, memberA.roles, Version.Zero) 51 val memberBWeaklyUp = new Member(memberB.uniqueAddress, Int.MaxValue, WeaklyUp, memberB.roles, Version.Zero) </pre>	
main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 42 (Code Correctness: Constructor Invokes Overridable Function)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: FunctionCall: cc Enclosing Method: SplitBrainResolverSettings() File: main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:42 Taint Flags:	
<pre> 39 40 private val cc = config.getConfig("akka.cluster.split-brain-resolver") 41 42 val DownningStableAfter: FiniteDuration = { 43 val key = "stable-after" 44 FiniteDuration(cc.getDuration(key).toMillis, TimeUnit.MILLISECONDS).requiring(_ >= Duration.Zero, key + " >= 0s") 45 } </pre>	



Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 48 (Code Correctness: Constructor Invokes Overridable Function)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cc**Enclosing Method:** SplitBrainResolverSettings()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:48**Taint Flags:**

```
45 }  
46  
47 val DowningStrategy: String =  
48 cc.getString("active-strategy").toLowerCase(Locale.ROOT) match {  
49 case strategyName if allStrategyNames(strategyName) => strategyName  
50 case unknown =>  
51 throw new ConfigurationException(
```

main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 57 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: cc**Enclosing Method:** SplitBrainResolverSettings()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:57**Taint Flags:**

```
54  
55 val DownAllWhenUnstable: FiniteDuration = {  
56 val key = "down-all-when-unstable"  
57 Helpers.toRootLowerCase(cc.getString("down-all-when-unstable")) match {  
58 case "on" =>  
59 // based on stable-after  
60 4.seconds.max(DowningStableAfter * 3 / 4)
```

main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 66 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Constructor Invokes Overridable Function**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 66 (Code Correctness: Constructor Invokes Overridable Function)****Low****Sink Details****Sink:** FunctionCall: cc**Enclosing Method:** SplitBrainResolverSettings()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:66**Taint Flags:**

```
63 Duration.Zero
64 case _ =>
65 FiniteDuration(cc.getDuration(key).toMillis, TimeUnit.MILLISECONDS)
66 .requiring(_ > Duration.Zero, key + " > 0s, or 'off' to disable")
67 }
68 }
69
```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 102 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: selfDc**Enclosing Method:** SplitBrainResolver()**File:** main/scala/akka/cluster/sbr/SplitBrainResolver.scala:102**Taint Flags:**

```
99
100 private val cluster = Cluster(context.system)
101
102 log.info(
103 s"SBR started. Config: strategy [{ }], stable-after [{ }], down-all-when-unstable [{ }], selfUniqueAddress [{ }], selfDc [$selfDc].",
104 Logging.simpleName(strategy.getClass),
105 stableAfter.toCoarsest,
```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 102 (Code Correctness: Constructor Invokes Overridable Function)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** FunctionCall: selfUniqueAddress**Enclosing Method:** SplitBrainResolver()

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.sbr	
main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 102 (Code Correctness: Constructor Invokes Overridable Function)	Low

File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:102

Taint Flags:

99

100 private val cluster = Cluster(context.system)

101

102 log.info(

103 s"SBR started. Config: strategy [{ }], stable-after [{ }], down-all-when-unstable [{ }], selfUniqueAddress [{ }], selfDc [\$selfDc].",

104 Logging.simpleName(strategy.getClass),

105 stableAfter.toCoarsest,

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 102 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: selfUniqueAddress

Enclosing Method: SplitBrainResolver()

File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:102

Taint Flags:

99

100 private val cluster = Cluster(context.system)

101

102 log.info(

103 s"SBR started. Config: strategy [{ }], stable-after [{ }], down-all-when-unstable [{ }], selfUniqueAddress [{ }], selfDc [\$selfDc].",

104 Logging.simpleName(strategy.getClass),

105 stableAfter.toCoarsest,

test/scala/akka/cluster/sbr/TestAddresses.scala, line 50 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: memberA

Enclosing Method: TestAddresses()

File: test/scala/akka/cluster/sbr/TestAddresses.scala:50

Taint Flags:

47 val memberG =



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

Package: akka.cluster.sbr

test/scala/akka/cluster/sbr/TestAddresses.scala, line 50 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

```

48 new Member(UniqueAddress(addressA.copy(host = Some("g")), 0L), 6, Up, Set(defaultDcRole), Version.Zero)
49
50 val memberAWeaklyUp = new Member(memberA.uniqueAddress, Int.MaxValue, WeaklyUp, memberA.roles, Version.Zero)
51 val memberBWeaklyUp = new Member(memberB.uniqueAddress, Int.MaxValue, WeaklyUp, memberB.roles, Version.Zero)
52
53 def dcMember(dc: ClusterSettings.DataCenter, m: Member): Member =

```

test/scala/akka/cluster/sbr/TestAddresses.scala, line 50 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: memberA
Enclosing Method: TestAddresses()
File: test/scala/akka/cluster/sbr/TestAddresses.scala:50
Taint Flags:

```

47 val memberG =
48 new Member(UniqueAddress(addressA.copy(host = Some("g")), 0L), 6, Up, Set(defaultDcRole), Version.Zero)
49
50 val memberAWeaklyUp = new Member(memberA.uniqueAddress, Int.MaxValue, WeaklyUp, memberA.roles, Version.Zero)
51 val memberBWeaklyUp = new Member(memberB.uniqueAddress, Int.MaxValue, WeaklyUp, memberB.roles, Version.Zero)
52
53 def dcMember(dc: ClusterSettings.DataCenter, m: Member): Member =

```

test/scala/akka/cluster/sbr/LeaseMajoritySpec.scala, line 19 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: default
Enclosing Method: LeaseMajoritySpec()
File: test/scala/akka/cluster/sbr/LeaseMajoritySpec.scala:19
Taint Flags:

```

16 akka.cluster.split-brain-resolver.lease-majority.lease-implementation = "akka.coordination.lease.kubernetes"
17 """)
18 .withFallback(ConfigFactory.load())
19 val blank = ConfigFactory.parseString("")

```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/LeaseMajoritySpec.scala, line 19 (Code Correctness: Constructor Invokes Overridable Function)	Low

```

20 akka.cluster.split-brain-resolver.lease-majority {
21   lease-name = " "
22 }""").withFallback(default)

```

test/scala/akka/cluster/sbr/LeaseMajoritySpec.scala, line 23 (Code Correctness: Constructor Invokes Overridable Function)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: default
Enclosing Method: LeaseMajoritySpec()
File: test/scala/akka/cluster/sbr/LeaseMajoritySpec.scala:23
Taint Flags:

```

20 akka.cluster.split-brain-resolver.lease-majority {
21   lease-name = " "
22 }""").withFallback(default)
23 val named = ConfigFactory.parseString("""
24 akka.cluster.split-brain-resolver.lease-majority {
25   lease-name = "shopping-cart-akka-sbr"
26 }""").withFallback(default)

```

Package: akka.cluster.testkit	
test/scala/akka/cluster/testkit/AutoDown.scala, line 64 (Code Correctness: Constructor Invokes Overridable Function)	Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: clusterSettings
Enclosing Method: AutoDowning()
File: test/scala/akka/cluster/testkit/AutoDown.scala:64
Taint Flags:

```

61 private val AutoDownUnreachableAfter: Duration = {
62   val key = "akka.cluster.testkit.auto-down-unreachable-after"
63   // it's not in reference.conf, since only used in tests
64   if (clusterSettings.config.hasPath(key)) {
65     toRootLowerCase(clusterSettings.config.getString(key)) match {
66       case "off" => Duration.Undefined

```



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

Package: akka.cluster.testkit

test/scala/akka/cluster/testkit/AutoDown.scala, line 64 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

```
67 case _ => clusterSettings.config.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off")
```

test/scala/akka/cluster/testkit/AutoDown.scala, line 65 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: clusterSettings

Enclosing Method: AutoDowning()

File: test/scala/akka/cluster/testkit/AutoDown.scala:65

Taint Flags:

```
62 val key = "akka.cluster.testkit.auto-down-unreachable-after"
63 // it's not in reference.conf, since only used in tests
64 if (clusterSettings.config.hasPath(key)) {
65   toRootLowerCase(clusterSettings.config.getString(key)) match {
66     case "off" => Duration.Undefined
67     case _ => clusterSettings.config.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off")
68   }
```

test/scala/akka/cluster/testkit/AutoDown.scala, line 67 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: clusterSettings

Enclosing Method: AutoDowning()

File: test/scala/akka/cluster/testkit/AutoDown.scala:67

Taint Flags:

```
64 if (clusterSettings.config.hasPath(key)) {
65   toRootLowerCase(clusterSettings.config.getString(key)) match {
66     case "off" => Duration.Undefined
67     case _ => clusterSettings.config.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off")
68   }
69 } else
70   Duration.Undefined
```



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.testkit	
test/scala/akka/cluster/testkit/AutoDownSpec.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: AutoDownSpec()
File: test/scala/akka/cluster/testkit/AutoDownSpec.scala:51
Taint Flags:

```

48 if (RARP(system).provider.remoteSettings.Artery.Enabled) "akka"
49 else "akka.tcp"
50
51 val memberA = TestMember(Address(protocol, "sys", "a", 2552), Up)
52 val memberB = TestMember(Address(protocol, "sys", "b", 2552), Up)
53 val memberC = TestMember(Address(protocol, "sys", "c", 2552), Up)
54

```

test/scala/akka/cluster/testkit/AutoDownSpec.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: AutoDownSpec()
File: test/scala/akka/cluster/testkit/AutoDownSpec.scala:52
Taint Flags:

```

49 else "akka.tcp"
50
51 val memberA = TestMember(Address(protocol, "sys", "a", 2552), Up)
52 val memberB = TestMember(Address(protocol, "sys", "b", 2552), Up)
53 val memberC = TestMember(Address(protocol, "sys", "c", 2552), Up)
54
55 def autoDownActor(autoDownUnreachableAfter: FiniteDuration): ActorRef =

```

test/scala/akka/cluster/testkit/AutoDownSpec.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)



Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.testkit	
test/scala/akka/cluster/testkit/AutoDownSpec.scala, line 53 (Code Correctness: Constructor Invokes Overridable Function)	Low
Sink Details	

Sink: FunctionCall: protocol
Enclosing Method: AutoDownSpec()
File: test/scala/akka/cluster/testkit/AutoDownSpec.scala:53
Taint Flags:

50
51 val memberA = TestMember(Address(protocol, "sys", "a", 2552), Up)
52 val memberB = TestMember(Address(protocol, "sys", "b", 2552), Up)
53 val memberC = TestMember(Address(protocol, "sys", "c", 2552), Up)
54
55 def autoDownActor(autoDownUnreachableAfter: FiniteDuration): ActorRef =
56 system.actorOf(Props(classOf[AutoDownTestActor], memberA, autoDownUnreachableAfter, testActor))

Code Correctness: Erroneous String Compare (19 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");
}
```

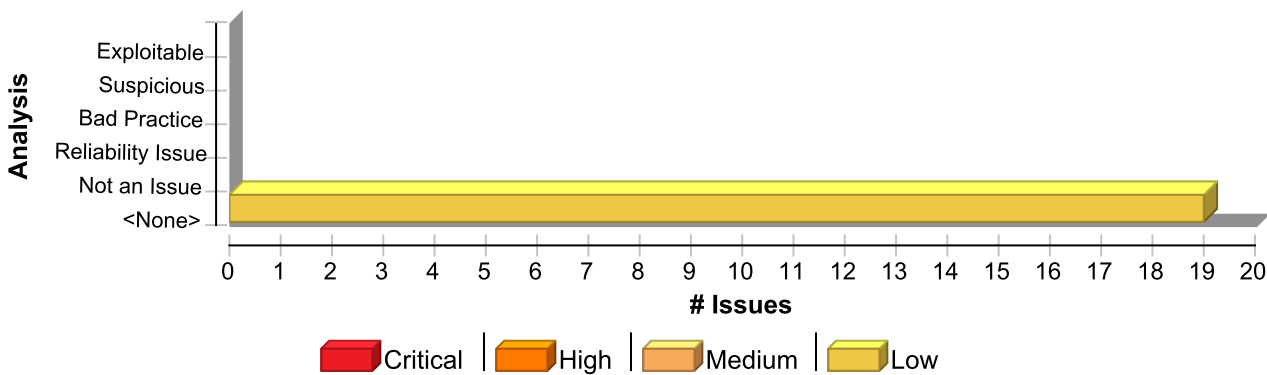
The `==` and `!=` operators will only behave as expected when they are used to compare strings contained in objects that are equal. The most common way for this to occur is for the strings to be interned, whereby the strings are added to a pool of objects maintained by the `String` class. Once a string 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 by 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	19	0	0	19
Total	19	0	0	19



Code Correctness: Erroneous String Compare**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 108 (Code Correctness: Erroneous String Compare)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:108**Taint Flags:**

```
105 val UnreachableNodesReaperInterval: FiniteDuration = cc.getMillisDuration("unreachable-nodes-reaper-interval")
106 val PublishStatsInterval: Duration = {
107   val key = "publish-stats-interval"
108   toRootLowerCase(cc.getString(key)) match {
109     case "off" => Duration.Undefined
110     case _ => cc.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off")
111   }
```

main/scala/akka/cluster/ClusterSettings.scala, line 124 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:124**Taint Flags:**

```
121
122 val DownRemovalMargin: FiniteDuration = {
123   val key = "down-removal-margin"
124   toRootLowerCase(cc.getString(key)) match {
125     case "off" => Duration.Zero
126     case _ => cc.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off")
127   }
```

main/scala/akka/cluster/ClusterSettings.scala, line 87 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Erroneous String Compare**Low****Package:** akka.cluster**main/scala/akka/cluster/ClusterSettings.scala, line 87 (Code Correctness: Erroneous String Compare)****Low****Sink Details****Sink:** Operation**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:87**Taint Flags:**

```
84 val SeedNodeTimeout: FiniteDuration = cc.getMillisDuration("seed-node-timeout")
85 val RetryUnsuccessfulJoinAfter: Duration = {
86 val key = "retry-unsuccessful-join-after"
87 toRootLowerCase(cc.getString(key)) match {
88 case "off" => Duration.Undefined
89 case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")
90 }
```

main/scala/akka/cluster/Cluster.scala, line 49 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** Cluster()**File:** main/scala/akka/cluster/Cluster.scala:49**Taint Flags:**

```
46 * INTERNAL API
47 */
48 private[cluster] final val isAssertInvariantsEnabled: Boolean =
49 System.getProperty("akka.cluster.assert", "off").toLowerCase match {
50 case "on" | "true" => true
51 case _ => false
52 }
```

main/scala/akka/cluster/Cluster.scala, line 49 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** Cluster()

Code Correctness: Erroneous String Compare**Low****Package:** akka.cluster**main/scala/akka/cluster/Cluster.scala, line 49 (Code Correctness: Erroneous String Compare)****Low****File:** main/scala/akka/cluster/Cluster.scala:49**Taint Flags:**

```
46 * INTERNAL API
47 */
48 private[cluster] final val isAssertInvariantsEnabled: Boolean =
49 System.getProperty("akka.cluster.assert", "off").toLowerCase match {
50 case "on" | "true" => true
51 case _ => false
52 }
```

main/scala/akka/cluster/ClusterSettings.scala, line 142 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:142**Taint Flags:**

```
139
140 val WeaklyUpAfter: FiniteDuration = {
141 val key = "allow-weakly-up-members"
142 toRootLowerCase(cc.getString(key)) match {
143 case "off" => Duration.Zero
144 case "on" => 7.seconds // for backwards compatibility when it wasn't a duration
145 case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")
}
```

main/scala/akka/cluster/ClusterSettings.scala, line 142 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** ClusterSettings()**File:** main/scala/akka/cluster/ClusterSettings.scala:142**Taint Flags:**

139



Code Correctness: Erroneous String Compare	Low
---	------------

Package: akka.cluster

main/scala/akka/cluster/ClusterSettings.scala, line 142 (Code Correctness: Erroneous String Compare)	Low
---	------------

```

140 val WeaklyUpAfter: FiniteDuration = {
141   val key = "allow-weakly-up-members"
142   toRootLowerCase(cc.getString(key)) match {
143     case "off" => Duration.Zero
144     case "on" => 7.seconds // for backwards compatibility when it wasn't a duration
145     case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")

```

main/scala/akka/cluster/ClusterSettings.scala, line 94 (Code Correctness: Erroneous String Compare)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:94
Taint Flags:

```

91 }
92 val ShutdownAfterUnsuccessfulJoinSeedNodes: Duration = {
93   val key = "shutdown-after-unsuccessful-join-seed-nodes"
94   toRootLowerCase(cc.getString(key)) match {
95     case "off" => Duration.Undefined
96     case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")
97 }

```

Package: akka.cluster.routing

main/scala/akka/cluster/routing/ClusterRouterConfig.scala, line 314 (Code Correctness: Erroneous String Compare)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: useRoleOption()
File: main/scala/akka/cluster/routing/ClusterRouterConfig.scala:314
Taint Flags:

```

311 * INTERNAL API
312 */
313 private[akka] object ClusterRouterSettingsBase {

```



Code Correctness: Erroneous String Compare**Low****Package:** akka.cluster.routing**main/scala/akka/cluster/routing/ClusterRouterConfig.scala, line 314 (Code Correctness: Erroneous String Compare)****Low**

```
314 def useRoleOption(role: String): Option[String] = role match {  
315   case null | "" => None  
316   case _ => Some(role)  
317 }
```

Package: akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 113 (Code Correctness: Erroneous String Compare)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** role()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:113**Taint Flags:**

```
110  
111 private def strategyConfig(strategyName: String): Config = cc.getConfig(strategyName)  
112  
113 private def role(c: Config): Option[String] = c.getString("role") match {  
114   case "" => None  
115   case r => Some(r)  
116 }
```

main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 42 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** downingActorProps()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala:42**Taint Flags:**

```
39 val cluster = Cluster(system)  
40 val selfDc = cluster.selfDataCenter  
41 val strategy =  
42 settings.DowningStrategy match {  
43   case KeepMajorityName =>
```



Code Correctness: Erroneous String Compare**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 42 (Code Correctness: Erroneous String Compare)****Low**

44 new KeepMajority(selfDc, settings.keepMajorityRole, cluster.selfUniqueAddress)

45 case StaticQuorumName =>

main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 57 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** SplitBrainResolverSettings()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:57**Taint Flags:**

54

55 val DownAllWhenUnstable: FiniteDuration = {

56 val key = "down-all-when-unstable"

57 Helpers.toRootLowerCase(cc.getString("down-all-when-unstable")) match {

58 case "on" =>

59 // based on stable-after

60 4.seconds.max(DowningStableAfter * 3 / 4)

main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 42 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** downingActorProps()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala:42**Taint Flags:**

39 val cluster = Cluster(system)

40 val selfDc = cluster.selfDataCenter

41 val strategy =

42 settings.DowningStrategy match {

43 case KeepMajorityName =>

44 new KeepMajority(selfDc, settings.keepMajorityRole, cluster.selfUniqueAddress)

45 case StaticQuorumName =>



Code Correctness: Erroneous String Compare**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 99 (Code Correctness: Erroneous String Compare)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** leaseMajoritySettings()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:99**Taint Flags:**

```
96 val acquireLeaseDelayForMinority =  
97 FiniteDuration(c.getDuration("acquire-lease-delay-for-minority").toMillis, TimeUnit.MILLISECONDS)  
98  
99 val leaseName = c.getString("lease-name").trim match {  
100 case "" => None  
101 case name => Some(name)  
102 }
```

main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 42 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** downingActorProps()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala:42**Taint Flags:**

```
39 val cluster = Cluster(system)  
40 val selfDc = cluster.selfDataCenter  
41 val strategy =  
42 settings.DowningStrategy match {  
43 case KeepMajorityName =>  
44 new KeepMajority(selfDc, settings.keepMajorityRole, cluster.selfUniqueAddress)  
45 case StaticQuorumName =>
```

main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 57 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)

Code Correctness: Erroneous String Compare**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 57 (Code Correctness: Erroneous String Compare)****Low****Sink Details****Sink:** Operation**Enclosing Method:** SplitBrainResolverSettings()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:57**Taint Flags:**

```
54
55 val DownAllWhenUnstable: FiniteDuration = {
56 val key = "down-all-when-unstable"
57 Helpers.toRootLowerCase(cc.getString("down-all-when-unstable")) match {
58 case "on" =>
59 // based on stable-after
60 4.seconds.max(DowningStableAfter * 3 / 4)
```

main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 42 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** downingActorProps()**File:** main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala:42**Taint Flags:**

```
39 val cluster = Cluster(system)
40 val selfDc = cluster.selfDataCenter
41 val strategy =
42 settings.DowningStrategy match {
43 case KeepMajorityName =>
44 new KeepMajority(selfDc, settings.keepMajorityRole, cluster.selfUniqueAddress)
45 case StaticQuorumName =>
```

main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 42 (Code Correctness: Erroneous String Compare)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** downingActorProps()

Code Correctness: Erroneous String Compare**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 42 (Code Correctness: Erroneous String Compare)****Low****File:** main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala:42**Taint Flags:**

```
39 val cluster = Cluster(system)
40 val selfDc = cluster.selfDataCenter
41 val strategy =
42 settings.DowningStrategy match {
43 case KeepMajorityName =>
44 new KeepMajority(selfDc, settings.keepMajorityRole, cluster.selfUniqueAddress)
45 case StaticQuorumName =>
```

Package: akka.cluster.testkit**test/scala/akka/cluster/testkit/AutoDown.scala, line 65 (Code Correctness: Erroneous String Compare)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Operation**Enclosing Method:** AutoDowning()**File:** test/scala/akka/cluster/testkit/AutoDown.scala:65**Taint Flags:**

```
62 val key = "akka.cluster.testkit.auto-down-unreachable-after"
63 // it's not in reference.conf, since only used in tests
64 if (clusterSettings.config.hasPath(key)) {
65 toRootLowerCase(clusterSettings.config.getString(key)) match {
66 case "off" => Duration.Undefined
67 case _ => clusterSettings.config.getMillisDuration(key).requiring(_ >= Duration.Zero, key + " >= 0s, or off")
68 }
```



Code Correctness: Non-Static Inner Class Implements Serializable (19 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`.

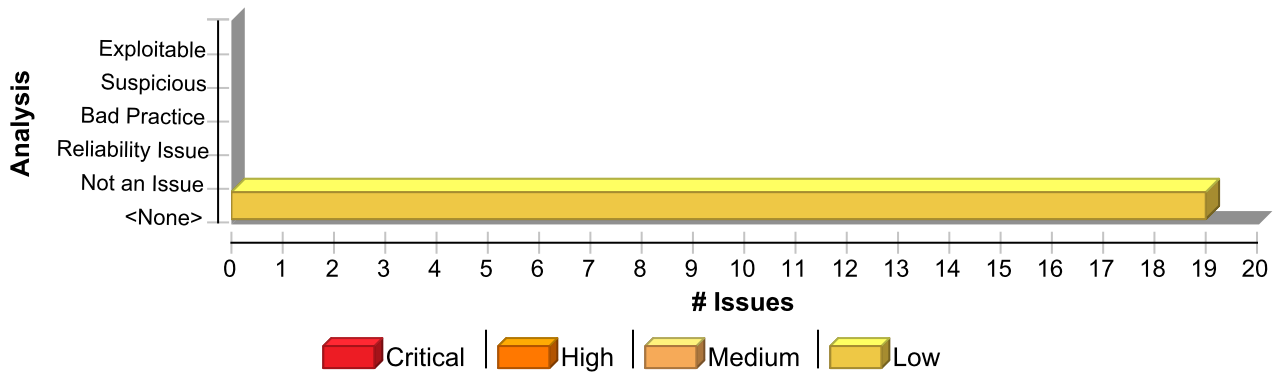
```
...
class User implements Serializable {
    private int accessLevel;
    class Registrator {
        ...
    }
}
```

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





Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Non-Static Inner Class Implements Serializable	19	0	0	19
Total	19	0	0	19

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

Package: akka.cluster

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 227 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: CrossDcHeartbeatSender\$MonitoringActive
File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:227
Taint Flags:

224
225 sealed trait StatusReport extends NoSerializationVerificationNeeded
226 sealed trait MonitoringStateReport extends StatusReport
227 final case class MonitoringActive(state: CrossDcHeartbeatingState) extends MonitoringStateReport
228 final case class MonitoringDormant() extends MonitoringStateReport
229 // -- end of messages intended only for local messaging during testing --
230 }

main/scala/akka/cluster/ClusterRemoteWatcher.scala, line 43 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterRemoteWatcher\$DelayedQuarantine



Code Correctness: Non-Static Inner Class Implements Serializable	Low
Package: akka.cluster	
main/scala/akka/cluster/ClusterRemoteWatcher.scala, line 43 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low

File: main/scala/akka/cluster/ClusterRemoteWatcher.scala:43

Taint Flags:

```

40 .withDispatcher(Dispatchers.InternalDispatcherId)
41 .withDeploy(Deploy.local)
42
43 private final case class DelayedQuarantine(m: Member, previousStatus: MemberStatus)
44 extends NoSerializationVerificationNeeded
45
46 }
```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 74 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterHeartbeatSender\$Heartbeat

File: main/scala/akka/cluster/ClusterHeartbeat.scala:74

Taint Flags:

```

71 /**
72  * Sent at regular intervals for failure detection.
73  */
74 final case class Heartbeat(from: Address, sequenceNr: Long, creationTimeNanos: Long)
75 extends ClusterMessage
76 with HeartbeatMessage
77 with DeadLetterSuppression
```

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 228 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: Class: CrossDcHeartbeatSender\$MonitoringDormant

File: main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:228

Taint Flags:

```

225 sealed trait StatusReport extends NoSerializationVerificationNeeded
226 sealed trait MonitoringStateReport extends StatusReport
227 final case class MonitoringActive(state: CrossDcHeartbeatingState) extends MonitoringStateReport
```



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

Package: akka.cluster

main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 228 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

```

228 final case class MonitoringDormant() extends MonitoringStateReport
229 // -- end of messages intended only for local messaging during testing --
230 }
231

```

test/scala/akka/cluster/StartupWithOneThreadSpec.scala, line 32 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: StartupWithOneThreadSpec\$GossipTo
File: test/scala/akka/cluster/StartupWithOneThreadSpec.scala:32
Taint Flags:

```

29 akka.actor.internal-dispatcher = akka.actor.default-dispatcher
30 ""
31
32 final case class GossipTo(address: Address)
33
34 def testProps =
35 Props(new Actor with ActorLogging {

```

main/scala/akka/cluster/ClusterHeartbeat.scala, line 82 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterHeartbeatSender\$HeartbeatRsp
File: main/scala/akka/cluster/ClusterHeartbeat.scala:82
Taint Flags:

```

79 /**
80 * Sent as reply to [[Heartbeat]] messages.
81 */
82 final case class HeartbeatRsp(from: UniqueAddress, sequenceNr: Long, creationTimeNanos: Long)
83 extends ClusterMessage
84 with HeartbeatMessage
85 with DeadLetterSuppression

```



Code Correctness: Non-Static Inner Class Implements Serializable**Low****Package:** akka.cluster**main/scala/akka/cluster/CrossDcClusterHeartbeat.scala, line 223 (Code Correctness: Non-Static Inner Class Implements Serializable)****Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Class: CrossDcHeartbeatSender\$ReportStatus**File:** main/scala/akka/cluster/CrossDcClusterHeartbeat.scala:223**Taint Flags:**

```
220
221 // -- messages intended only for local messaging during testing --
222 sealed trait InspectionCommand extends NoSerializationVerificationNeeded
223 final case class ReportStatus()
224
225 sealed trait StatusReport extends NoSerializationVerificationNeeded
226 sealed trait MonitoringStateReport extends StatusReport
```

main/scala/akka/cluster/Reachability.scala, line 28 (Code Correctness: Non-Static Inner Class Implements Serializable)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Class: Reachability\$Record**File:** main/scala/akka/cluster/Reachability.scala:28**Taint Flags:**

```
25 }
26
27 @SerialVersionUID(1L)
28 final case class Record(observer: UniqueAddress, subject: UniqueAddress, status: ReachabilityStatus, version: Long)
29
30 sealed trait ReachabilityStatus
31 @SerialVersionUID(1L) case object Reachable extends ReachabilityStatus
```

test/scala/akka/cluster/ClusterSpec.scala, line 48 (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	
test/scala/akka/cluster/ClusterSpec.scala, line 48 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low

Sink: Class: ClusterSpec\$GossipTo
File: test/scala/akka/cluster/ClusterSpec.scala:48
Taint Flags:

```

45 akka.remote.artery.canonical.port = 0
46 ""
47
48 final case class GossipTo(address: Address)
49 }
50
51 class ClusterSpec extends AkkaSpec(ClusterSpec.config) with ImplicitSender {

```

main/scala/akka/cluster/ClusterHeartbeat.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: ClusterHeartbeatSender\$ExpectedFirstHeartbeat
File: main/scala/akka/cluster/ClusterHeartbeat.scala:89
Taint Flags:

```

86
87 // sent to self only
88 case object HeartbeatTick
89 final case class ExpectedFirstHeartbeat(from: UniqueAddress)
90
91 }
92

```

main/scala/akka/cluster/ClusterReadView.scala, line 27 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
--	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterReadView\$State
File: main/scala/akka/cluster/ClusterReadView.scala:27
Taint Flags:

```

24 * INTERNAL API
25 */

```



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

Package: akka.cluster

main/scala/akka/cluster/ClusterReadView.scala, line 27 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
--	------------

```

26 @InternalApi private[akka] object ClusterReadView {
27   final case class State(
28     clusterState: CurrentClusterState,
29     reachability: Reachability,
30     selfMember: Member,

```

Package: akka.cluster.sbr

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 67 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: SplitBrainResolver\$ReachabilityChangedStats
File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:67
Taint Flags:

```

64 final case class WhenTimeElapsed(deadline: Deadline) extends ReleaseLeaseCondition
65 }
66
67 final case class ReachabilityChangedStats(
68   firstChangeTimestamp: Long,
69   latestChangeTimestamp: Long,
70   changeCount: Long) {

```

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 38 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: SplitBrainResolverSpec\$DownCalled
File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:38
Taint Flags:

```

35
36 object SplitBrainResolverSpec {
37
38   final case class DownCalled(address: Address)
39
40   object DowningTestActor {

```



Code Correctness: Non-Static Inner Class Implements Serializable	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 38 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
41 def props(
main/scala/akka/cluster/sbr/DowningStrategy.scala, line 42 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: Class: DowningStrategy\$AcquireLeaseAndDownUnreachable File: main/scala/akka/cluster/sbr/DowningStrategy.scala:42 Taint Flags:	
39 sealed trait AcquireLeaseDecision extends Decision { 40 def acquireDelay: FiniteDuration 41 } 42 final case class AcquireLeaseAndDownUnreachable(acquireDelay: FiniteDuration) extends AcquireLeaseDecision { 43 override def isIndirectlyConnected = false 44 } 45 final case class AcquireLeaseAndDownIndirectlyConnected(acquireDelay: FiniteDuration) extends AcquireLeaseDecision {	
main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 48 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
Issue Details	
Kingdom: Code Quality Scan Engine: SCA (Structural)	
Sink Details	
Sink: Class: SplitBrainResolver\$AcquireLeaseResult File: main/scala/akka/cluster/sbr/SplitBrainResolver.scala:48 Taint Flags:	
45 /** 46 * Response (result) of the acquire lease request. 47 */ 48 final case class AcquireLeaseResult(holdingLease: Boolean) 49 50 /** 51 * Response (result) of the release lease request.	
main/scala/akka/cluster/sbr/DowningStrategy.scala, line 45 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
Issue Details	



Code Correctness: Non-Static Inner Class Implements Serializable**Low****Package:** akka.cluster.sbr**main/scala/akka/cluster/sbr/DowningStrategy.scala, line 45 (Code Correctness: Non-Static Inner Class Implements Serializable)****Low****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Class: DowningStrategy\$AcquireLeaseAndDownIndirectlyConnected**File:** main/scala/akka/cluster/sbr/DowningStrategy.scala:45**Taint Flags:**

```
42 final case class AcquireLeaseAndDownUnreachable(acquireDelay: FiniteDuration) extends AcquireLeaseDecision {  
43   override def isIndirectlyConnected = false  
44 }  
45 final case class AcquireLeaseAndDownIndirectlyConnected(acquireDelay: FiniteDuration) extends AcquireLeaseDecision {  
46   override def isIndirectlyConnected = true  
47 }  
48 case object ReverseDownIndirectlyConnected extends Decision {
```

main/scala/akka/cluster/sbr/SplitBrainResolver.scala, line 53 (Code Correctness: Non-Static Inner Class Implements Serializable)**Low****Issue Details****Kingdom:** Code Quality**Scan Engine:** SCA (Structural)**Sink Details****Sink:** Class: SplitBrainResolver\$ReleaseLeaseResult**File:** main/scala/akka/cluster/sbr/SplitBrainResolver.scala:53**Taint Flags:**

```
50 /**  
51  * Response (result) of the release lease request.  
52  */  
53 final case class ReleaseLeaseResult(released: Boolean)  
54  
55 /**  
56  * For delayed acquire of the lease.
```

Package: akka.cluster.testkit**test/scala/akka/cluster/testkit/AutoDown.scala, line 90 (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.testkit	
test/scala/akka/cluster/testkit/AutoDown.scala, line 90 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low

Sink: Class: AutoDown\$UnreachableTimeout
File: test/scala/akka/cluster/testkit/AutoDown.scala:90
Taint Flags:

```

87 def props(autoDownUnreachableAfter: FiniteDuration): Props =
88   Props(classOf[AutoDown], autoDownUnreachableAfter)
89
90 final case class UnreachableTimeout(node: UniqueAddress)
91 }
92
93 /**

```

test/scala/akka/cluster/testkit/AutoDownSpec.scala, line 22 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low
---	------------

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: AutoDownSpec\$DownCalled
File: test/scala/akka/cluster/testkit/AutoDownSpec.scala:22
Taint Flags:

```

19 import akka.testkit.TimingTest
20
21 object AutoDownSpec {
22   final case class DownCalled(address: Address)
23
24   class AutoDownTestActor(memberA: Member, autoDownUnreachableAfter: FiniteDuration, probe: ActorRef)
25     extends AutoDownBase(autoDownUnreachableAfter) {

```



Dead Code: Expression is Always false (1 issue)

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();
    }
}
```

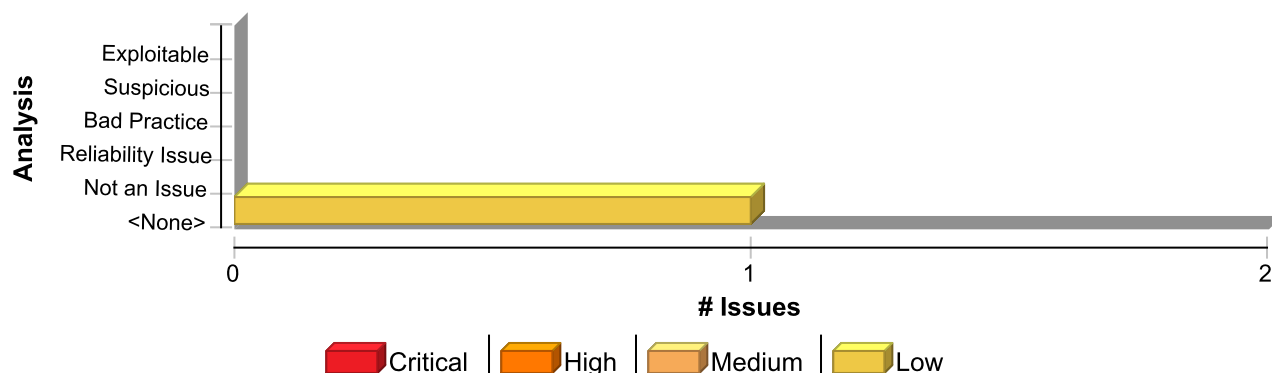
Recommendation

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



contributing to the functionality of the program.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Dead Code: Expression is Always false	1	0	0	1
Total	1	0	0	1

Dead Code: Expression is Always false

Low

Package: akka.cluster.routing

test/scala/akka/cluster/routing/ClusterRouterSupervisorSpec.scala, line 17 (Dead Code: Expression is Always false)

Low

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: IfStatement

Enclosing Method: applyOrElse()

File: test/scala/akka/cluster/routing/ClusterRouterSupervisorSpec.scala:17

Taint Flags:

```
14 class KillableActor() extends Actor {  
15  
16 def receive = {  
17 case "go away" =>  
18 throw new IllegalArgumentException("Goodbye then!")  
19 }  
20
```



Insecure Randomness (6 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");  
}
```

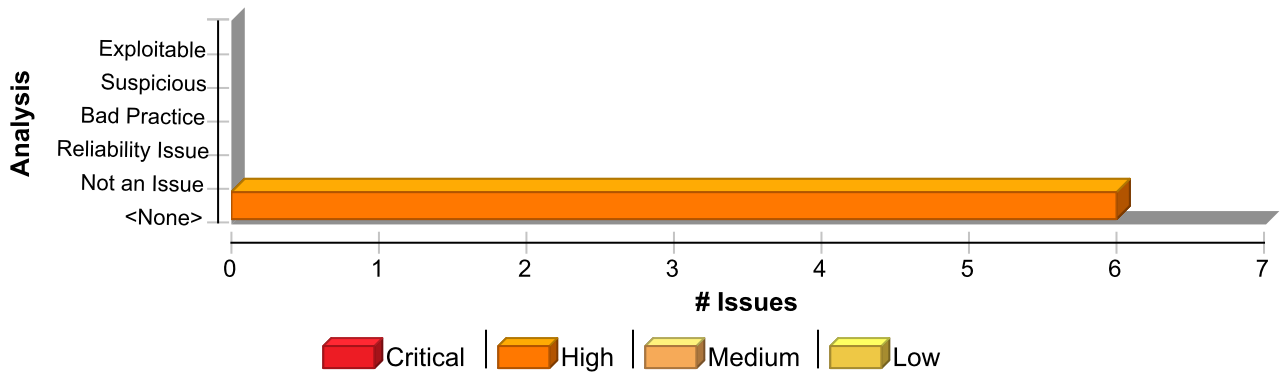
This code uses the `Random.nextInt()` function to generate "unique" identifiers for the receipt pages it generates. Since `Random.nextInt()` is a statistical PRNG, it is easy for an attacker to guess the strings it generates. Although the underlying design of the receipt system is also faulty, it would 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





Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Insecure Randomness	6	0	0	6
Total	6	0	0	6

Insecure Randomness **High**

Package: akka.cluster

main/scala/akka/cluster/MembershipState.scala, line 404 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextDouble()
Enclosing Method: preferNodesWithDifferentView()
File: main/scala/akka/cluster/MembershipState.scala:404
Taint Flags:

```

401 }
402
403 protected def preferNodesWithDifferentView(state: MembershipState): Boolean =
404   ThreadLocalRandom.current.nextDouble() < adjustedGossipDifferentViewProbability(state.latestGossip.members.size)
405
406 protected def dcsInRandomOrder(dcs: List[DataCenter]): List[DataCenter] =
407   Random.shuffle(dcs)

```

main/scala/akka/cluster/MembershipState.scala, line 411 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextInt()
Enclosing Method: selectRandomNode()
File: main/scala/akka/cluster/MembershipState.scala:411
Taint Flags:



Insecure Randomness	High
Package: akka.cluster	
main/scala/akka/cluster/MembershipState.scala, line 411 (Insecure Randomness)	High

```

408
409 protected def selectRandomNode(nodes: IndexedSeq[UniqueAddress]): Option[UniqueAddress] =
410 if (nodes.isEmpty) None
411 else Some(nodes(ThreadLocalRandom.current.nextInt(nodes.size)))
412 }
413
414 undefined

```

test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala, line 158 (Insecure Randomness)	High
--	-------------

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextInt()
Enclosing Method: rndNode()
File: test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala:158
Taint Flags:

```

155 val rnd = ThreadLocalRandom.current
156 val nodes =
157 (1 to rnd.nextInt(10, 200)).map(n => UniqueAddress(Address("akka", "sys", "n" + n, 2552), n.toLong)).toVector
158 def rndNode() = nodes(rnd.nextInt(0, nodes.size))
159 val selfUniqueAddress = rndNode()
160 var state = emptyState(selfUniqueAddress)
161 val Add = 0

```

main/scala/akka/cluster/MembershipState.scala, line 400 (Insecure Randomness)	High
--	-------------

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextDouble()
Enclosing Method: selectDcLocalNodes()
File: main/scala/akka/cluster/MembershipState.scala:400
Taint Flags:

```

397 // don't go below the configured probability
398 math.max((5 - localMembers) * 0.25, crossDcGossipProbability)
399 }
400 ThreadLocalRandom.current.nextDouble() > probability
401 }

```



Insecure Randomness	High
Package: akka.cluster	
main/scala/akka/cluster/MembershipState.scala, line 400 (Insecure Randomness)	High

```

402
403 protected def preferNodesWithDifferentView(state: MembershipState): Boolean =

```

Package: test.scala.akka.cluster	
test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala, line 156 (Insecure Randomness)	High

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextInt()
Enclosing Method: apply()
File: test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala:156
Taint Flags:

```

153
154 "behave correctly for random operations" in {
155   val rnd = ThreadLocalRandom.current
156   val nodes =
157     (1 to rnd.nextInt(10, 200)).map(n => UniqueAddress(Address("akka", "sys", "n" + n, 2552), n.toLong)).toVector
158   def rndNode() = nodes(rnd.nextInt(0, nodes.size))
159   val selfUniqueAddress = rndNode()

```

test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala, line 166 (Insecure Randomness)	High
--	-------------

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextInt()
Enclosing Method: apply()
File: test/scala/akka/cluster/ClusterHeartbeatSenderStateSpec.scala:166
Taint Flags:

```

163 val Unreachable = 2
164 val HeartbeatRsp = 3
165 for (i <- 1 to 100000) {
166   val operation = rnd.nextInt(Add, HeartbeatRsp + 1)
167   val node = rndNode()
168   try {
169     operation match {

```



J2EE Bad Practices: Threads (13 issues)

Abstract

Thread management in a web application is forbidden in some circumstances and is always highly error prone.

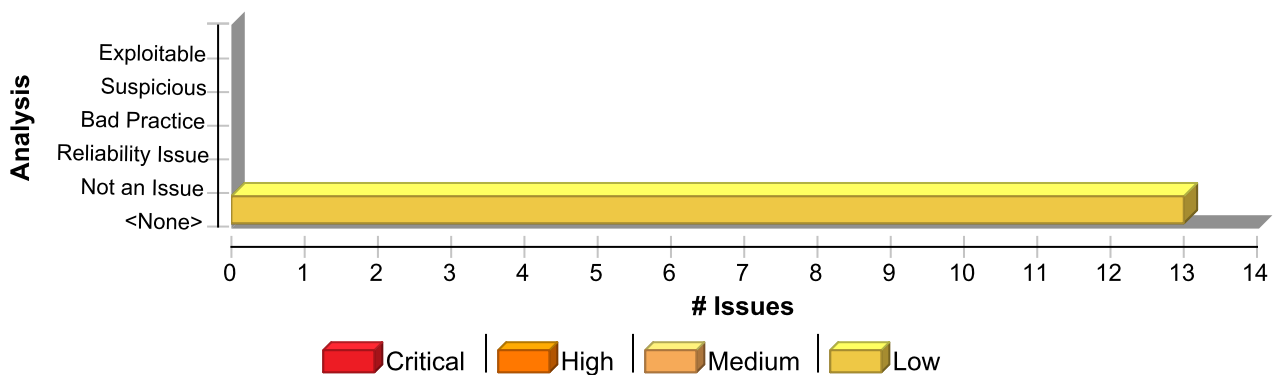
Explanation

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	13	0	0	13
Total	13	0	0	13

J2EE Bad Practices: Threads Low

Package: akka.cluster

main/scala/akka/cluster/Cluster.scala, line 423 (J2EE Bad Practices: Threads) Low

Issue Details

Kingdom: Time and State

Scan Engine: SCA (Semantic)

Sink Details

Sink: run()

Enclosing Method: registerOnMemberRemoved()

File: main/scala/akka/cluster/Cluster.scala:423

Taint Flags:



J2EE Bad Practices: Threads	Low
------------------------------------	------------

Package: akka.cluster

main/scala/akka/cluster/Cluster.scala, line 423 (J2EE Bad Practices: Threads)	Low
--	------------

```

420 */
421 def registerOnMemberRemoved(callback: Runnable): Unit = {
422   if (_isTerminated.get())
423     callback.run()
424   else
425     clusterDaemons ! InternalClusterAction.AddOnMemberRemovedListener(callback)
426 }
```

main/scala/akka/cluster/ClusterDaemon.scala, line 1666 (J2EE Bad Practices: Threads)	Low
---	------------

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: run()
Enclosing Method: akka\$cluster\$OnMemberStatusChangeListener\$\$done()
File: main/scala/akka/cluster/ClusterDaemon.scala:1666
Taint Flags:

```

1663 }
1664
1665 private def done(): Unit = {
1666   try callback.run()
1667   catch {
1668     case NonFatal(e) => logError(e, "[{}] callback failed with [{}]", s"On${to.getSimpleName}", e.getMessage)
1669   } finally {
```

Package: akka.cluster.sbr

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1657 (J2EE Bad Practices: Threads)	Low
--	------------

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: sleep()
Enclosing Method: SplitBrainResolverSpec\$\$anon\$93()
File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1657
Taint Flags:

```

1654 tick()
1655 expectNoDecision(100.millis)
1656
1657 Thread.sleep(3000)
```



J2EE Bad Practices: Threads		Low
Package: akka.cluster.sbr		
test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1657 (J2EE Bad Practices: Threads)		Low
<pre> 1658 tick() 1659 expectNoDecision(100.millis) 1660 }</pre>		
test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1621 (J2EE Bad Practices: Threads)		Low
Issue Details		
Kingdom: Time and State Scan Engine: SCA (Semantic)		
Sink Details		
Sink: sleep() Enclosing Method: SplitBrainResolverSpec\$\$anon\$92() File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1621 Taint Flags:		
<pre> 1618 expectNoDecision(100.millis) 1619 // 1800 ms has elapsed 1620 1621 Thread.sleep(1000) 1622 // E and D are still unreachable 1623 reachabilityChanged(memberA -> memberE, memberB -> memberD) 1624 tick()</pre>		
test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1641 (J2EE Bad Practices: Threads)		Low
Issue Details		
Kingdom: Time and State Scan Engine: SCA (Semantic)		
Sink Details		
Sink: sleep() Enclosing Method: SplitBrainResolverSpec\$\$anon\$93() File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1641 Taint Flags:		
<pre> 1638 tick() 1639 expectNoDecision(100.millis) 1640 1641 Thread.sleep(1000) 1642 reachabilityChanged(memberB -> memberD) 1643 reachable(memberE) 1644 tick()</pre>		



J2EE Bad Practices: Threads	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1580 (J2EE Bad Practices: Threads)	Low

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: sleep()
Enclosing Method: SplitBrainResolverSpec\$\$\$anon\$91()
File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1580
Taint Flags:

```

1577 tick()
1578 expectNoDecision(100.millis)
1579
1580 Thread.sleep(1000)
1581 reachabilityChanged(memberB -> memberD)
1582 reachable(memberE)
1583 tick()

```

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1575 (J2EE Bad Practices: Threads)	Low
--	------------

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: sleep()
Enclosing Method: SplitBrainResolverSpec\$\$\$anon\$91()
File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1575
Taint Flags:

```

1572 tick()
1573 expectNoDecision(100.millis)
1574
1575 Thread.sleep(1000)
1576 reachabilityChanged(memberB -> memberD, memberB -> memberE)
1577 tick()
1578 expectNoDecision(100.millis)

```

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1569 (J2EE Bad Practices: Threads)	Low
--	------------

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)



J2EE Bad Practices: Threads	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1569 (J2EE Bad Practices: Threads)	Low

Sink Details

Sink: sleep()
Enclosing Method: SplitBrainResolverSpec\$\$anon\$91()
File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1569
Taint Flags:

```

1566 tick()
1567 expectNoDecision(100.millis)
1568
1569 Thread.sleep(1000)
1570 reachabilityChanged(memberB -> memberD)
1571 reachable(memberE)
1572 tick()

```

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1614 (J2EE Bad Practices: Threads)	Low
--	------------

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: sleep()
Enclosing Method: SplitBrainResolverSpec\$\$anon\$92()
File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1614
Taint Flags:

```

1611 expectNoDecision(100.millis)
1612 // 1200 ms has elapsed
1613
1614 Thread.sleep(500)
1615 // E and D are unreachable, reset stableDeadline
1616 reachabilityChanged(memberA -> memberE, memberB -> memberD, memberC -> memberD)
1617 tick()

```

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1607 (J2EE Bad Practices: Threads)	Low
--	------------

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: sleep()
Enclosing Method: SplitBrainResolverSpec\$\$anon\$92()



J2EE Bad Practices: Threads	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1607 (J2EE Bad Practices: Threads)	Low

File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1607

Taint Flags:

```

1604 expectNoDecision(100.millis)
1605 // 600 ms has elapsed
1606
1607 Thread.sleep(500)
1608 reachabilityChanged(memberA -> memberE)
1609 reachable(memberD) // reset stableDeadline
1610 tick()
```

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1600 (J2EE Bad Practices: Threads)	Low
--	------------

Issue Details

Kingdom: Time and State

Scan Engine: SCA (Semantic)

Sink Details

Sink: sleep()

Enclosing Method: SplitBrainResolverSpec\$\$anon\$92()

File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1600

Taint Flags:

```

1597 tick()
1598 expectNoDecision(100.millis)
1599
1600 Thread.sleep(500)
1601 // E and D are still unreachable
1602 reachabilityChanged(memberA -> memberE, memberB -> memberD)
1603 tick()
```

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1653 (J2EE Bad Practices: Threads)	Low
--	------------

Issue Details

Kingdom: Time and State

Scan Engine: SCA (Semantic)

Sink Details

Sink: sleep()

Enclosing Method: SplitBrainResolverSpec\$\$anon\$93()

File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1653

Taint Flags:

```

1650 expectNoDecision(100.millis)
```



J2EE Bad Practices: Threads	Low
Package: akka.cluster.sbr	
test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1653 (J2EE Bad Practices: Threads)	Low

```

1651 reachabilityChanged()
1652 reachable(memberD)
1653 Thread.sleep(500)
1654 tick()
1655 expectNoDecision(100.millis)
1656

```

test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala, line 1648 (J2EE Bad Practices: Threads)	Low
--	------------

Issue Details

Kingdom: Time and State
Scan Engine: SCA (Semantic)

Sink Details

Sink: sleep()
Enclosing Method: SplitBrainResolverSpec\$\$anon\$93()
File: test/scala/akka/cluster/sbr/SplitBrainResolverSpec.scala:1648
Taint Flags:

```

1645 expectNoDecision(100.millis)
1646
1647 // wait longer than stableAfter
1648 Thread.sleep(500)
1649 tick()
1650 expectNoDecision(100.millis)
1651 reachabilityChanged()

```



Key Management: Hardcoded Encryption Key (6 issues)

Abstract

Hardcoded encryption keys can compromise security in a way that cannot be easily remedied.

Explanation

It is never a good idea to hardcode an encryption key because it allows all of the project's developers to view the encryption key, and makes fixing the problem extremely difficult. After the code is in production, a software patch is required to change the encryption key. If the account that is protected by the encryption key is compromised, the owners of the system must choose between security and availability. **Example 1:** The following code uses a hardcoded encryption key:

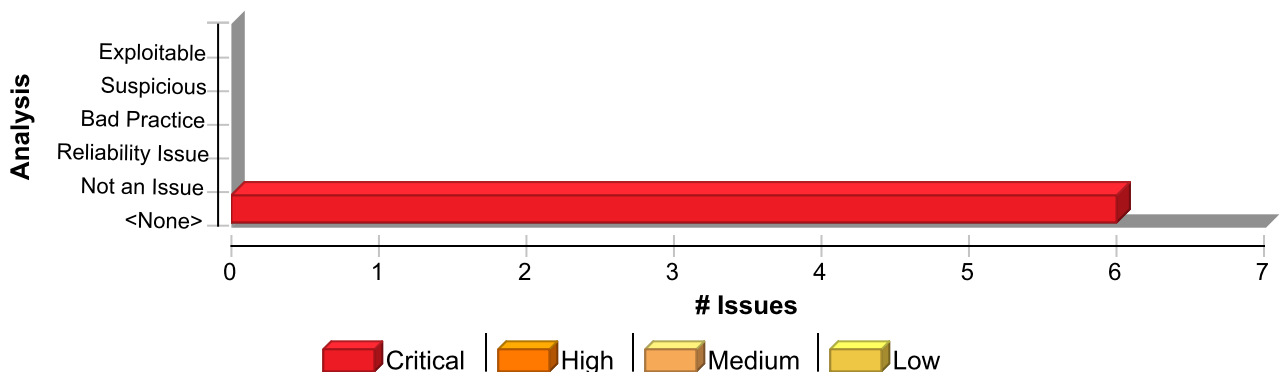
```
...
private static final String encryptionKey = "lakdsljkalkjlkksdfkl";
byte[] keyBytes = encryptionKey.getBytes();
SecretKeySpec key = new SecretKeySpec(keyBytes, "AES");
Cipher encryptCipher = Cipher.getInstance("AES");
encryptCipher.init(Cipher.ENCRYPT_MODE, key);
...
```

Anyone with access to the code has access to the encryption key. After the application has shipped, there is no way to change the encryption key unless the program is patched. An employee with access to this information can use it to break into the system. If attackers had access to the executable for the application, they could extract the encryption key value.

Recommendation

Encryption keys should never be hardcoded and should be obfuscated and managed in an external source. Storing encryption keys in plain text anywhere on the system allows anyone with sufficient permissions to read and potentially misuse the encryption key.

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Key Management: Hardcoded Encryption Key	6	0	0	6
Total	6	0	0	6



Key Management: Hardcoded Encryption Key	Critical
Package: akka.cluster	
main/scala/akka/cluster/ClusterSettings.scala, line 86 (Key Management: Hardcoded Encryption Key)	Critical
Issue Details	

Kingdom: Security Features
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: key
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:86
Taint Flags:

```

83 }.toVector
84 val SeedNodeTimeout: FiniteDuration = cc.getMillisDuration("seed-node-timeout")
85 val RetryUnsuccessfulJoinAfter: Duration = {
86   val key = "retry-unsuccessful-join-after"
87   toRootLowerCase(cc.getString(key)) match {
88     case "off" => Duration.Undefined
89     case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")

```

main/scala/akka/cluster/ClusterSettings.scala, line 86 (Key Management: Hardcoded Encryption Key)	Critical
Issue Details	

Kingdom: Security Features
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: key
Enclosing Method: ClusterSettings()
File: main/scala/akka/cluster/ClusterSettings.scala:86
Taint Flags:

```

83 }.toVector
84 val SeedNodeTimeout: FiniteDuration = cc.getMillisDuration("seed-node-timeout")
85 val RetryUnsuccessfulJoinAfter: Duration = {
86   val key = "retry-unsuccessful-join-after"
87   toRootLowerCase(cc.getString(key)) match {
88     case "off" => Duration.Undefined
89     case _ => cc.getMillisDuration(key).requiring(_ > Duration.Zero, key + " > 0s, or off")

```

Package: akka.cluster.sbr	
main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 43 (Key Management: Hardcoded Encryption Key)	Critical
Issue Details	

Kingdom: Security Features



Key Management: Hardcoded Encryption Key	Critical
Package: akka.cluster.sbr	
main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 43 (Key Management: Hardcoded Encryption Key)	Critical

Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: key
Enclosing Method: SplitBrainResolverSettings()
File: main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:43
Taint Flags:

```

40 private val cc = config.getConfig("akka.cluster.split-brain-resolver")
41
42 val DownStableAfter: FiniteDuration = {
43 val key = "stable-after"
44 FiniteDuration(cc.getDuration(key).toMillis, TimeUnit.MILLISECONDS).requiring(_ >= Duration.Zero, key + " >= 0s")
45 }
46

```

main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala, line 43 (Key Management: Hardcoded Encryption Key)	Critical
---	-----------------

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: key
Enclosing Method: SplitBrainResolverSettings()
File: main/scala/akka/cluster/sbr/SplitBrainResolverSettings.scala:43
Taint Flags:

```

40 private val cc = config.getConfig("akka.cluster.split-brain-resolver")
41
42 val DownStableAfter: FiniteDuration = {
43 val key = "stable-after"
44 FiniteDuration(cc.getDuration(key).toMillis, TimeUnit.MILLISECONDS).requiring(_ >= Duration.Zero, key + " >= 0s")
45 }
46

```

Package: akka.cluster.testkit	
test/scala/akka/cluster/testkit/AutoDown.scala, line 62 (Key Management: Hardcoded Encryption Key)	Critical

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Structural)



Key Management: Hardcoded Encryption Key	Critical
Package: akka.cluster.testkit	
test/scala/akka/cluster/testkit/AutoDown.scala, line 62 (Key Management: Hardcoded Encryption Key)	Critical
Sink Details	

Sink: VariableAccess: key
Enclosing Method: AutoDowning()
File: test/scala/akka/cluster/testkit/AutoDown.scala:62
Taint Flags:

```

59 private def clusterSettings = Cluster(system).settings
60
61 private val AutoDownUnreachableAfter: Duration = {
62   val key = "akka.cluster.testkit.auto-down-unreachable-after"
63   // it's not in reference.conf, since only used in tests
64   if (clusterSettings.config.hasPath(key)) {
65     toRootLowerCase(clusterSettings.config.getString(key)) match {

```

test/scala/akka/cluster/testkit/AutoDown.scala, line 62 (Key Management: Hardcoded Encryption Key)	Critical
Issue Details	

Kingdom: Security Features
Scan Engine: SCA (Structural)

Sink Details	
Sink: VariableAccess: key Enclosing Method: AutoDowning() File: test/scala/akka/cluster/testkit/AutoDown.scala:62 Taint Flags:	
<pre> 59 private def clusterSettings = Cluster(system).settings 60 61 private val AutoDownUnreachableAfter: Duration = { 62 val key = "akka.cluster.testkit.auto-down-unreachable-after" 63 // it's not in reference.conf, since only used in tests 64 if (clusterSettings.config.hasPath(key)) { 65 toRootLowerCase(clusterSettings.config.getString(key)) match { </pre>	



Poor Style: Value Never Read (2 issues)

Abstract

The variable's value is assigned but never used, making it a dead store.

Explanation

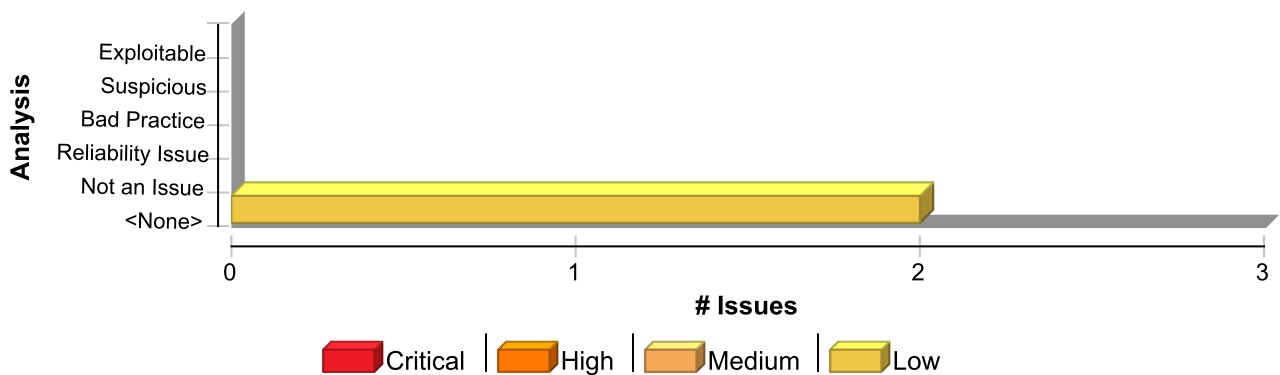
This variable's value is not used. After the assignment, the variable is either assigned another value or goes out of scope. **Example:** The following code excerpt assigns to the variable `r` and then overwrites the value without using it.

```
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	2	0	0	2
Total	2	0	0	2

Poor Style: Value Never Read

Low

Package: akka.cluster.sbr

main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 40 (Poor Style: Value Never Read)

Low

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: selfDc

Enclosing Method: downingActorProps()

File: main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala:40



Poor Style: Value Never Read	Low
Package: akka.cluster.sbr	
main/scala/akka/cluster/sbr/SplitBrainResolverProvider.scala, line 40 (Poor Style: Value Never Read)	Low

Taint Flags:

```

37 import SplitBrainResolverSettings._
38
39 val cluster = Cluster(system)
40 val selfDc = cluster.selfDataCenter
41 val strategy =
42 settings.DowningStrategy match {
43 case KeepMajorityName =>

```

Package: test.scala.akka.cluster.protobuf	
test/scala/akka/cluster/protobuf/ClusterMessageSerializerSpec.scala, line 154 (Poor Style: Value Never Read)	Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: VariableAccess: reachability3
Enclosing Method: apply()
File: test/scala/akka/cluster/protobuf/ClusterMessageSerializerSpec.scala:154
Taint Flags:

```

151 val node4 = VectorClock.Node("node4")
152 val g1 = (Gossip(SortedSet(a1, b1, c1, d1)) :+ node1 :+ node2).seen(a1.uniqueAddress).seen(b1.uniqueAddress)
153 val g2 = (g1 :+ node3 :+ node4).seen(a1.uniqueAddress).seen(c1.uniqueAddress)
154 val reachability3 = Reachability.empty
155 .unreachable(a1.uniqueAddress, e1.uniqueAddress)
156 .unreachable(b1.uniqueAddress, e1.uniqueAddress)
157 checkDeserializationWithManifest(

```



Weak Cryptographic Hash (2 issues)

Abstract

Weak cryptographic hashes cannot guarantee data integrity and should not be used in security-critical contexts.

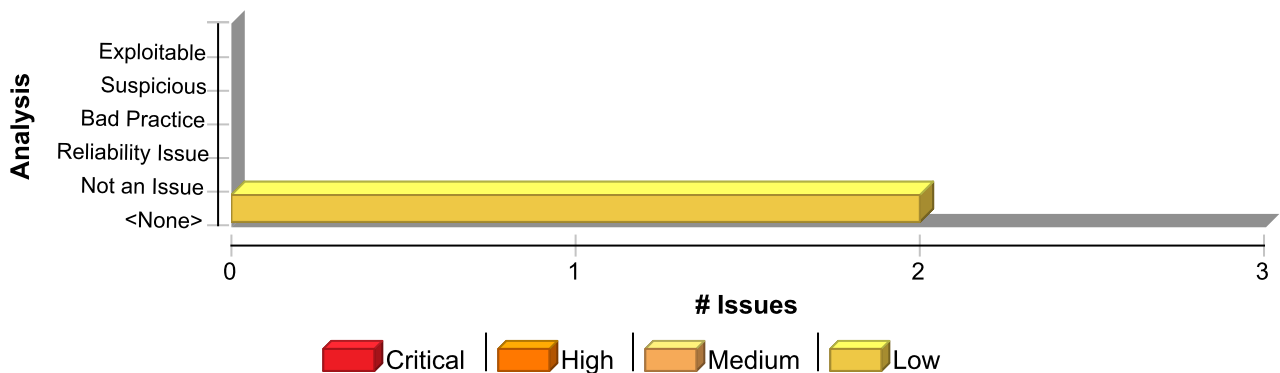
Explanation

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	2	0	0	2
Total	2	0	0	2

Weak Cryptographic Hash	Low
Package: akka.cluster	
main/scala/akka/cluster/VectorClock.scala, line 31 (Weak Cryptographic Hash)	Low
Issue Details	

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details



Weak Cryptographic Hash	Low
Package: akka.cluster	
main/scala/akka/cluster/VectorClock.scala, line 31 (Weak Cryptographic Hash)	Low

Sink: getInstance()
Enclosing Method: hash()
File: main/scala/akka/cluster/VectorClock.scala:31
Taint Flags:

```

28 def fromHash(hash: String): Node = hash
29
30 private def hash(name: String): String = {
31   val digester = MessageDigest.getInstance("MD5")
32   digester.update(name.getBytes("UTF-8"))
33   digester.digest.map { h =>
34     "%02x".format(0xFF & h)

```

main/scala/akka/cluster/Gossip.scala, line 288 (Weak Cryptographic Hash)	Low
Issue Details	

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: getInstance()
Enclosing Method: seenDigest\$lzycompute()
File: main/scala/akka/cluster/Gossip.scala:288
Taint Flags:

```

285 seen: Set[UniqueAddress] = Set.empty,
286 reachability: Reachability = Reachability.empty) {
287
288   lazy val seenDigest: Array[Byte] = {
289     val bytes = seen.toVector.sorted.map(node => node.address).mkString(",").getBytes(StandardCharsets.UTF_8)
290     MessageDigest.getInstance("SHA-1").digest(bytes)
291   }

```



