Fortify Standalone Report Generator

Developer Workbook

akka-cluster-tools

Table of Contents

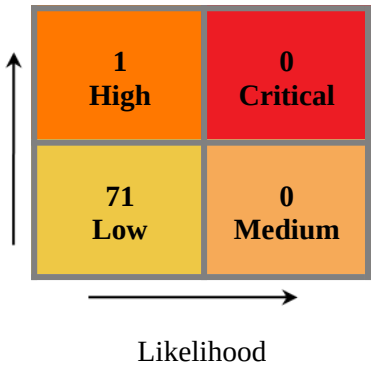
Executive Summary

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-tools 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-tools
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
WebInspect Agent:	Results Not Present
Other:	Results Not Present

Issues by Priority



Top Ten Critical Categories

This project does not contain any critical issues

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, 11:24 AM	**Engine Version:**	21.1.1.0009
Host Name:	Jacks-Work-MBP.local	**Certification:**	VALID
Number of Files:	8	**Lines of Code:**	1,874

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: Constructor Invokes Overridable Function	0	0	0	0 / 48	0 / 48
Code Correctness: Erroneous String Compare	0	0	0	0 / 6	0 / 6
Code Correctness: Non-Static Inner Class Implements Serializable	0	0	0	0 / 17	0 / 17
Insecure Randomness	0	0 / 1	0	0	0 / 1

Results Outline

Code Correctness: Constructor Invokes Overridable Function (48 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");
    }
  }
}
```

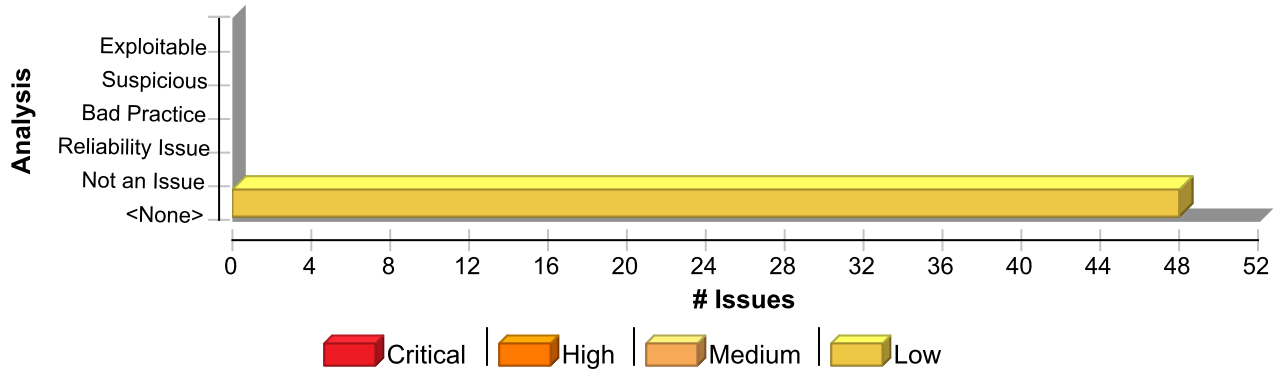
The code in `Example 1` prints "Attack successful!", since the `Attacker` class overrides the `validateUser()` function that is called from the constructor of the superclass `User`, and Java will first look in the subclass for functions called from the constructor.

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	48	0	0	48
Total	**48**	**0**	**0**	**48**

Code Correctness: Constructor Invokes Overridable Function	**Low**
Package: akka.cluster.client	
client/ClusterClient.scala, line 581 (Code Correctness: Constructor Invokes Overridable Function)	**Low**
Issue Details	

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.client

client/ClusterClient.scala, line 581 (Code Correctness: Constructor Invokes Overridable Function) — Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: config
Enclosing Method: ClusterClientReceptionist()
File: client/ClusterClient.scala:581
Taint Flags:

```
578 final class ClusterClientReceptionist(system: ExtendedActorSystem) extends Extension {
579
580  private val config = system.settings.config.getConfig("akka.cluster.client.receptionist")
581  private val role: Option[String] = config.getString("role") match {
582  case "" => None
583  case r => Some(r)
584  }
```

client/ClusterClient.scala, line 634 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: config
Enclosing Method: ClusterClientReceptionist()
File: client/ClusterClient.scala:634
Taint Flags:

```
631  private val receptionist: ActorRef = {
632  if (isTerminated)
633  system.deadLetters
634  else {
635  val name = config.getString("name")
636  val dispatcher = config.getString("use-dispatcher")
637  // important to use val mediator here to activate it outside of ClusterReceptionist constructor
```

client/ClusterClient.scala, line 635 (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.client

client/ClusterClient.scala, line 635 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: config
Enclosing Method: ClusterClientReceptionist()
File: client/ClusterClient.scala:635
Taint Flags:

```
632 if (isTerminated)
633 system.deadLetters
634 else {
635 val name = config.getString("name")
636 val dispatcher = config.getString("use-dispatcher")
637 // important to use val mediator here to activate it outside of ClusterReceptionist constructor
638 val mediator = pubSubMediator
```

client/ClusterClient.scala, line 636 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: config
Enclosing Method: ClusterClientReceptionist()
File: client/ClusterClient.scala:636
Taint Flags:

```
633 system.deadLetters
634 else {
635 val name = config.getString("name")
636 val dispatcher = config.getString("use-dispatcher")
637 // important to use val mediator here to activate it outside of ClusterReceptionist constructor
638 val mediator = pubSubMediator
639 system.systemActorOf(
```

client/ClusterClient.scala, line 391 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: initialContactsSel
Enclosing Method: ClusterClient()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.client

client/ClusterClient.scala, line 391 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: client/ClusterClient.scala:391
Taint Flags:

```
388 initialContacts.to(HashSet)
389 val initialContactsSel =
390 contactPaths.map(context.actorSelection)
391 var contacts = initialContactsSel
392 sendGetContacts()
393
394 var contactPathsPublished = contactPaths
```

client/ClusterClient.scala, line 632 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: isTerminated
Enclosing Method: ClusterClientReceptionist()
File: client/ClusterClient.scala:632
Taint Flags:

```
629 * The [[ClusterReceptionist]] actor
630 */
631 private val receptionist: ActorRef = {
632 if (isTerminated)
633 system.deadLetters
634 else {
635 val name = config.getString("name")
```

client/ClusterClient.scala, line 957 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterReceptionist()
File: client/ClusterClient.scala:957
Taint Flags:

```
954 import settings._
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.client

client/ClusterClient.scala, line 957 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
955
956 val cluster = Cluster(context.system)
957 val verboseHeartbeat = cluster.settings.Debug.VerboseHeartbeatLogging
958 import cluster.selfAddress
959
960 require(role.forall(cluster.selfRoles.contains), s"This cluster member [$selfAddress] doesn't have the role [$role]")
```

client/ClusterClient.scala, line 960 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterReceptionist()
File: client/ClusterClient.scala:960
Taint Flags:

```
957 val verboseHeartbeat = cluster.settings.Debug.VerboseHeartbeatLogging
958 import cluster.selfAddress
959
960 require(role.forall(cluster.selfRoles.contains), s"This cluster member [$selfAddress] doesn't have the role [$role]")
961
962 var nodes: immutable.SortedSet[Address] = {
963 def hashFor(node: Address): Int = node match {
```

client/ClusterClient.scala, line 389 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: contactPaths
Enclosing Method: ClusterClient()
File: client/ClusterClient.scala:389
Taint Flags:

```
386
387 var contactPaths: HashSet[ActorPath] =
388 initialContacts.to(HashSet)
389 val initialContactsSel =
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.client

client/ClusterClient.scala, line 389 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
390 contactPaths.map(context.actorSelection)
391 var contacts = initialContactsSel
392 sendGetContacts()
```

client/ClusterClient.scala, line 394 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: contactPaths
Enclosing Method: ClusterClient()
File: client/ClusterClient.scala:394
Taint Flags:

```
391 var contacts = initialContactsSel
392 sendGetContacts()
393
394 var contactPathsPublished = contactPaths
395
396 var subscribers = Vector.empty[ActorRef]
397
```

client/ClusterClient.scala, line 392 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: sendGetContacts
Enclosing Method: ClusterClient()
File: client/ClusterClient.scala:392
Taint Flags:

```
389 val initialContactsSel =
390 contactPaths.map(context.actorSelection)
391 var contacts = initialContactsSel
392 sendGetContacts()
393
394 var contactPathsPublished = contactPaths
395
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.client

client/ClusterClient.scala, line 761 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: _failureDetectionInterval_=
Enclosing Method: ClusterReceptionistSettings()
File: client/ClusterClient.scala:761
Taint Flags:

```
758 this(role, numberOfContacts, responseTunnelReceiveTimeout)
759 this._heartbeatInterval = heartbeatInterval
760 this._acceptableHeartbeatPause = acceptableHeartbeatPause
761 this._failureDetectionInterval = failureDetectionInterval
762 }
763
764 // END BINARY COMPATIBILITY
```

client/ClusterClient.scala, line 760 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: _acceptableHeartbeatPause_=
Enclosing Method: ClusterReceptionistSettings()
File: client/ClusterClient.scala:760
Taint Flags:

```
757 failureDetectionInterval: FiniteDuration) = {
758 this(role, numberOfContacts, responseTunnelReceiveTimeout)
759 this._heartbeatInterval = heartbeatInterval
760 this._acceptableHeartbeatPause = acceptableHeartbeatPause
761 this._failureDetectionInterval = failureDetectionInterval
762 }
763
```

client/ClusterClient.scala, line 759 (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.client

client/ClusterClient.scala, line 759 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: _heartbeatInterval_=
Enclosing Method: ClusterReceptionistSettings()
File: client/ClusterClient.scala:759
Taint Flags:

```
756 acceptableHeartbeatPause: FiniteDuration,
757 failureDetectionInterval: FiniteDuration) = {
758 this(role, numberOfContacts, responseTunnelReceiveTimeout)
759 this._heartbeatInterval = heartbeatInterval
760 this._acceptableHeartbeatPause = acceptableHeartbeatPause
761 this._failureDetectionInterval = failureDetectionInterval
762 }
```

client/ClusterClient.scala, line 638 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: pubSubMediator
Enclosing Method: ClusterClientReceptionist()
File: client/ClusterClient.scala:638
Taint Flags:

```
635 val name = config.getString("name")
636 val dispatcher = config.getString("use-dispatcher")
637 // important to use val mediator here to activate it outside of ClusterReceptionist constructor
638 val mediator = pubSubMediator
639 system.systemActorOf(
640 ClusterReceptionist.props(mediator, ClusterReceptionistSettings(config)).withDispatcher(dispatcher),
641 name)
```

client/ClusterClient.scala, line 402 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: scheduleRefreshContactsTick
Enclosing Method: ClusterClient()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.client

client/ClusterClient.scala, line 402 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: client/ClusterClient.scala:402
Taint Flags:

```
399 val heartbeatTask =
400 context.system.scheduler.scheduleWithFixedDelay(heartbeatInterval, heartbeatInterval, self, HeartbeatTick)
401 var refreshContactsTask: Option[Cancellable] = None
402 scheduleRefreshContactsTick(establishingGetContactsInterval)
403 self ! RefreshContactsTick
404
405 var buffer = MessageBuffer.empty
```

client/ClusterClient.scala, line 977 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: virtualNodesFactor
Enclosing Method: ClusterReceptionist()
File: client/ClusterClient.scala:977
Taint Flags:

```
974 immutable.SortedSet()
975 }
976 val virtualNodesFactor = 10
977 var consistentHash: ConsistentHash[Address] = ConsistentHash(nodes, virtualNodesFactor)
978
979 var clientInteractions = HashMap.empty[ActorRef, DeadlineFailureDetector]
980 var clientsPublished = HashSet.empty[ActorRef]
```

client/ClusterClient.scala, line 977 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: nodes
Enclosing Method: ClusterReceptionist()
File: client/ClusterClient.scala:977
Taint Flags:

```
974 immutable.SortedSet()
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.client

client/ClusterClient.scala, line 977 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
975 }
976 val virtualNodesFactor = 10
977 var consistentHash: ConsistentHash[Address] = ConsistentHash(nodes, virtualNodesFactor)
978
979 var clientInteractions = HashMap.empty[ActorRef, DeadlineFailureDetector]
980 var clientsPublished = HashSet.empty[ActorRef]
```

Package: akka.cluster.client.protobuf

client/protobuf/ClusterClientMessageSerializer.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetContactsManifest
Enclosing Method: ClusterClientMessageSerializer()
File: client/protobuf/ClusterClientMessageSerializer.scala:35
Taint Flags:

```
32
33 private val emptyByteArray = Array.empty[Byte]
34
35 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
36 ContactsManifest -> contactsFromBinary,
37 GetContactsManifest -> { _ =>
38 GetContacts
```

client/protobuf/ClusterClientMessageSerializer.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: HeartbeatManifest
Enclosing Method: ClusterClientMessageSerializer()
File: client/protobuf/ClusterClientMessageSerializer.scala:35
Taint Flags:

```
32
33 private val emptyByteArray = Array.empty[Byte]
34
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.client.protobuf

client/protobuf/ClusterClientMessageSerializer.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
35 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
36 ContactsManifest -> contactsFromBinary,
37 GetContactsManifest -> { _ =>
38 GetContacts
```

client/protobuf/ClusterClientMessageSerializer.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ContactsManifest
Enclosing Method: ClusterClientMessageSerializer()
File: client/protobuf/ClusterClientMessageSerializer.scala:35
Taint Flags:

```
32
33 private val emptyByteArray = Array.empty[Byte]
34
35 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
36 ContactsManifest -> contactsFromBinary,
37 GetContactsManifest -> { _ =>
38 GetContacts
```

client/protobuf/ClusterClientMessageSerializer.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ReceptionistShutdownManifest
Enclosing Method: ClusterClientMessageSerializer()
File: client/protobuf/ClusterClientMessageSerializer.scala:35
Taint Flags:

```
32
33 private val emptyByteArray = Array.empty[Byte]
34
35 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
36 ContactsManifest -> contactsFromBinary,
37 GetContactsManifest -> { _ =>
```

Code Correctness: Constructor Invokes Overridable Function	Low

Package: akka.cluster.client.protobuf

client/protobuf/ClusterClientMessageSerializer.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
38 GetContacts
```

client/protobuf/ClusterClientMessageSerializer.scala, line 35 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: HeartbeatRspManifest
Enclosing Method: ClusterClientMessageSerializer()
File: client/protobuf/ClusterClientMessageSerializer.scala:35
Taint Flags:

```
32
33 private val emptyByteArray = Array.empty[Byte]
34
35 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
36 ContactsManifest -> contactsFromBinary,
37 GetContactsManifest -> { _ =>
38 GetContacts
```

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 584 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: pruneInterval
Enclosing Method: DistributedPubSubMediator()
File: pubsub/DistributedPubSubMediator.scala:584
Taint Flags:

```
581 import context.dispatcher
582 val gossipTask = context.system.scheduler.scheduleWithFixedDelay(gossipInterval, gossipInterval, self, GossipTick)
583 val pruneInterval: FiniteDuration = removedTimeToLive / 2
584 val pruneTask = context.system.scheduler.scheduleWithFixedDelay(pruneInterval, pruneInterval, self, Prune)
585
586 var registry: Map[Address, Bucket] = Map.empty.withDefault(a => Bucket(a, 0L, TreeMap.empty))
587 var nodes: Set[Address] = Set.empty
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 584 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: pruneInterval
Enclosing Method: DistributedPubSubMediator()
File: pubsub/DistributedPubSubMediator.scala:584
Taint Flags:

```
581 import context.dispatcher
582 val gossipTask = context.system.scheduler.scheduleWithFixedDelay(gossipInterval, gossipInterval, self, GossipTick)
583 val pruneInterval: FiniteDuration = removedTimeToLive / 2
584 val pruneTask = context.system.scheduler.scheduleWithFixedDelay(pruneInterval, pruneInterval, self, Prune)
585
586 var registry: Map[Address, Bucket] = Map.empty.withDefault(a => Bucket(a, 0L, TreeMap.empty))
587 var nodes: Set[Address] = Set.empty
```

pubsub/DistributedPubSubMediator.scala, line 955 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: isTerminated
Enclosing Method: DistributedPubSub()
File: pubsub/DistributedPubSubMediator.scala:955
Taint Flags:

```
952 * The [[DistributedPubSubMediator]]
953 */
954 val mediator: ActorRef = {
955 if (isTerminated)
956 system.deadLetters
957 else {
958 val name = system.settings.config.getString("akka.cluster.pub-sub.name")
```

pubsub/DistributedPubSubMediator.scala, line 575 (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.pubsub

pubsub/DistributedPubSubMediator.scala, line 575 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: DistributedPubSubMediator()
File: pubsub/DistributedPubSubMediator.scala:575
Taint Flags:

```
572 import cluster.selfAddress
573
574 require(
575 role.forall(cluster.selfRoles.contains),
576 s"This cluster member [${selfAddress}] doesn't have the role [$role]")
577
578 val removedTimeToLiveMillis = removedTimeToLive.toMillis
```

pubsub/DistributedPubSubMediator.scala, line 957 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: settings
Enclosing Method: DistributedPubSub()
File: pubsub/DistributedPubSubMediator.scala:957
Taint Flags:

```
954 val mediator: ActorRef = {
955 if (isTerminated)
956 system.deadLetters
957 else {
958 val name = system.settings.config.getString("akka.cluster.pub-sub.name")
959 val dispatcher = system.settings.config.getString("akka.cluster.pub-sub.use-dispatcher")
960 system.systemActorOf(DistributedPubSubMediator.props(settings).withDispatcher(dispatcher), name)
```

Package: akka.cluster.pubsub.protobuf

pubsub/protobuf/DistributedPubSubMessageSerializer.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: SendToAllManifest

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.pubsub.protobuf

pubsub/protobuf/DistributedPubSubMessageSerializer.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — Low

Enclosing Method: DistributedPubSubMessageSerializer()
File: pubsub/protobuf/DistributedPubSubMessageSerializer.scala:43
Taint Flags:

```
40 private val PublishManifest = "E"
41 private val SendToOneSubscriberManifest = "F"
42
43 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
44 StatusManifest -> statusFromBinary,
45 DeltaManifest -> deltaFromBinary,
46 SendManifest -> sendFromBinary,
```

pubsub/protobuf/DistributedPubSubMessageSerializer.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: SendManifest
Enclosing Method: DistributedPubSubMessageSerializer()
File: pubsub/protobuf/DistributedPubSubMessageSerializer.scala:43
Taint Flags:

```
40 private val PublishManifest = "E"
41 private val SendToOneSubscriberManifest = "F"
42
43 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
44 StatusManifest -> statusFromBinary,
45 DeltaManifest -> deltaFromBinary,
46 SendManifest -> sendFromBinary,
```

pubsub/protobuf/DistributedPubSubMessageSerializer.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: PublishManifest
Enclosing Method: DistributedPubSubMessageSerializer()
File: pubsub/protobuf/DistributedPubSubMessageSerializer.scala:43
Taint Flags:

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.pubsub.protobuf

pubsub/protobuf/DistributedPubSubMessageSerializer.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
40  private val PublishManifest = "E"
41  private val SendToOneSubscriberManifest = "F"
42
43  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
44  StatusManifest -> statusFromBinary,
45  DeltaManifest -> deltaFromBinary,
46  SendManifest -> sendFromBinary,
```

pubsub/protobuf/DistributedPubSubMessageSerializer.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: SendToOneSubscriberManifest
Enclosing Method: DistributedPubSubMessageSerializer()
File: pubsub/protobuf/DistributedPubSubMessageSerializer.scala:43
Taint Flags:

```
40  private val PublishManifest = "E"
41  private val SendToOneSubscriberManifest = "F"
42
43  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
44  StatusManifest -> statusFromBinary,
45  DeltaManifest -> deltaFromBinary,
46  SendManifest -> sendFromBinary,
```

pubsub/protobuf/DistributedPubSubMessageSerializer.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: StatusManifest
Enclosing Method: DistributedPubSubMessageSerializer()
File: pubsub/protobuf/DistributedPubSubMessageSerializer.scala:43
Taint Flags:

```
40  private val PublishManifest = "E"
41  private val SendToOneSubscriberManifest = "F"
42
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.pubsub.protobuf

pubsub/protobuf/DistributedPubSubMessageSerializer.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — **Low**

```
43 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
44 StatusManifest -> statusFromBinary,
45 DeltaManifest -> deltaFromBinary,
46 SendManifest -> sendFromBinary,
```

pubsub/protobuf/DistributedPubSubMessageSerializer.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — **Low**

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: DeltaManifest
Enclosing Method: DistributedPubSubMessageSerializer()
File: pubsub/protobuf/DistributedPubSubMessageSerializer.scala:43
Taint Flags:

```
40 private val PublishManifest = "E"
41 private val SendToOneSubscriberManifest = "F"
42
43 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
44 StatusManifest -> statusFromBinary,
45 DeltaManifest -> deltaFromBinary,
46 SendManifest -> sendFromBinary,
```

Package: akka.cluster.singleton

singleton/ClusterSingletonManager.scala, line 545 (Code Correctness: Constructor Invokes Overridable Function) — **Low**

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: coordShutdown
Enclosing Method: ClusterSingletonManager()
File: singleton/ClusterSingletonManager.scala:545
Taint Flags:

```
542 // for CoordinatedShutdown
543 val coordShutdown = CoordinatedShutdown(context.system)
544 val memberExitingProgress = Promise[Done]()
545 coordShutdown.addTask(CoordinatedShutdown.PhaseClusterExiting, "wait-singleton-exiting") { () =>
546 if (cluster.isTerminated || cluster.selfMember.status == MemberStatus.Down)
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.singleton

singleton/ClusterSingletonManager.scala, line 545 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
547 Future.successful(Done)
548 else
```

singleton/ClusterSingletonManager.scala, line 551 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: coordShutdown
Enclosing Method: ClusterSingletonManager()
File: singleton/ClusterSingletonManager.scala:551
Taint Flags:

```
548 else
549 memberExitingProgress.future
550 }
551 coordShutdown.addTask(CoordinatedShutdown.PhaseClusterExiting, "singleton-exiting-2") { () =>
552 if (cluster.isTerminated || cluster.selfMember.status == MemberStatus.Down) {
553 Future.successful(Done)
554 } else {
```

singleton/ClusterSingletonProxy.scala, line 175 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ageOrdering
Enclosing Method: ClusterSingletonProxy()
File: singleton/ClusterSingletonProxy.scala:175
Taint Flags:

```
172 var singleton: Option[ActorRef] = None
173 // sort by age, oldest first
174 val ageOrdering = Member.ageOrdering
175 var membersByAge: immutable.SortedSet[Member] = immutable.SortedSet.empty(ageOrdering)
176
177 var buffer: MessageBuffer = MessageBuffer.empty
178
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.singleton

singleton/ClusterSingletonManager.scala, line 490 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterSingletonManager()
File: singleton/ClusterSingletonManager.scala:490
Taint Flags:

```
487 import settings._
488
489 val cluster = Cluster(context.system)
490 val selfUniqueAddressOption = Some(cluster.selfUniqueAddress)
491 import cluster.settings.LogInfo
492
493 require(
```

singleton/ClusterSingletonManager.scala, line 494 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterSingletonManager()
File: singleton/ClusterSingletonManager.scala:494
Taint Flags:

```
491 import cluster.settings.LogInfo
492
493 require(
494 role.forall(cluster.selfRoles.contains),
495 s"This cluster member [${cluster.selfAddress}] doesn't have the role [$role]")
496
497 private val singletonLeaseName = s"${context.system.name}-singleton-${self.path}"
```

singleton/ClusterSingletonManager.scala, line 511 (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.singleton

singleton/ClusterSingletonManager.scala, line 511 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterSingletonManager()
File: singleton/ClusterSingletonManager.scala:511
Taint Flags:

```
508 }
509
510 val removalMargin =
511 if (settings.removalMargin <= Duration.Zero) cluster.downingProvider.downRemovalMargin
512 else settings.removalMargin
513
514 val (maxHandOverRetries, maxTakeOverRetries) = {
```

singleton/ClusterSingletonManager.scala, line 515 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: removalMargin
Enclosing Method: ClusterSingletonManager()
File: singleton/ClusterSingletonManager.scala:515
Taint Flags:

```
512 else settings.removalMargin
513
514 val (maxHandOverRetries, maxTakeOverRetries) = {
515 val n = (removalMargin.toMillis / handOverRetryInterval.toMillis).toInt
516 val minRetries = context.system.settings.config.getInt("akka.cluster.singleton.min-number-of-hand-over-retries")
517 require(minRetries >= 1, "min-number-of-hand-over-retries must be >= 1")
518 val handOverRetries = math.max(minRetries, n + 3)
```

singleton/ClusterSingletonProxy.scala, line 167 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: createIdentifyId
Enclosing Method: ClusterSingletonProxy()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.singleton

singleton/ClusterSingletonProxy.scala, line 167 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: singleton/ClusterSingletonProxy.scala:167
Taint Flags:

```
164 import settings._
165 val singletonPath = (singletonManagerPath + "/" + settings.singletonName).split("/")
166 var identifyCounter = 0
167 var identifyId = createIdentifyId(identifyCounter)
168 def createIdentifyId(i: Int) = "identify-singleton-" + singletonPath.mkString("/") + i
169 var identifyTimer: Option[Cancellable] = None
170
```

singleton/ClusterSingletonProxy.scala, line 197 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ClusterSingletonProxy()
File: singleton/ClusterSingletonProxy.scala:197
Taint Flags:

```
194
195 private val targetDcRole = settings.dataCenter match {
196 case Some(t) => ClusterSettings.DcRolePrefix + t
197 case None => ClusterSettings.DcRolePrefix + cluster.settings.SelfDataCenter
198 }
199
200 def matchingRole(member: Member): Boolean =
```

singleton/ClusterSingletonProxy.scala, line 167 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: identifyCounter
Enclosing Method: ClusterSingletonProxy()
File: singleton/ClusterSingletonProxy.scala:167
Taint Flags:

```
164 import settings._
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.singleton

singleton/ClusterSingletonProxy.scala, line 167 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
165 val singletonPath = (singletonManagerPath + "/" + settings.singletonName).split("/")
166 var identifyCounter = 0
167 var identifyId = createIdentifyId(identifyCounter)
168 def createIdentifyId(i: Int) = "identify-singleton-" + singletonPath.mkString("/") + i
169 var identifyTimer: Option[Cancellable] = None
170
```

Package: akka.cluster.singleton.protobuf

singleton/protobuf/ClusterSingletonMessageSerializer.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: TakeOverFromMeManifest
Enclosing Method: ClusterSingletonMessageSerializer()
File: singleton/protobuf/ClusterSingletonMessageSerializer.scala:33
Taint Flags:

```
30
31 private val emptyByteArray = Array.empty[Byte]
32
33 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](HandOverToMeManifest -> { _ =>
34 HandOverToMe
35 }, HandOverInProgressManifest -> { _ =>
36 HandOverInProgress
```

singleton/protobuf/ClusterSingletonMessageSerializer.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: HandOverToMeManifest
Enclosing Method: ClusterSingletonMessageSerializer()
File: singleton/protobuf/ClusterSingletonMessageSerializer.scala:33
Taint Flags:

```
30
31 private val emptyByteArray = Array.empty[Byte]
32
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.singleton.protobuf

singleton/protobuf/ClusterSingletonMessageSerializer.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
33 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](HandOverToMeManifest -> { _ =>
34 HandOverToMe
35 }, HandOverInProgressManifest -> { _ =>
36 HandOverInProgress
```

singleton/protobuf/ClusterSingletonMessageSerializer.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: HandOverInProgressManifest
Enclosing Method: ClusterSingletonMessageSerializer()
File: singleton/protobuf/ClusterSingletonMessageSerializer.scala:33
Taint Flags:

```
30
31 private val emptyByteArray = Array.empty[Byte]
32
33 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](HandOverToMeManifest -> { _ =>
34 HandOverToMe
35 }, HandOverInProgressManifest -> { _ =>
36 HandOverInProgress
```

singleton/protobuf/ClusterSingletonMessageSerializer.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: HandOverDoneManifest
Enclosing Method: ClusterSingletonMessageSerializer()
File: singleton/protobuf/ClusterSingletonMessageSerializer.scala:33
Taint Flags:

```
30
31 private val emptyByteArray = Array.empty[Byte]
32
33 private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](HandOverToMeManifest -> { _ =>
34 HandOverToMe
35 }, HandOverInProgressManifest -> { _ =>
```

Code Correctness: Constructor Invokes Overridable Function	Low
Package: akka.cluster.singleton.protobuf	
singleton/protobuf/ClusterSingletonMessageSerializer.scala, line 33 (Code Correctness: Constructor Invokes Overridable Function)	**Low**

```
36  HandOverInProgress
```

Code Correctness: Erroneous String Compare (6 issues)

Abstract

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

Explanation

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

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

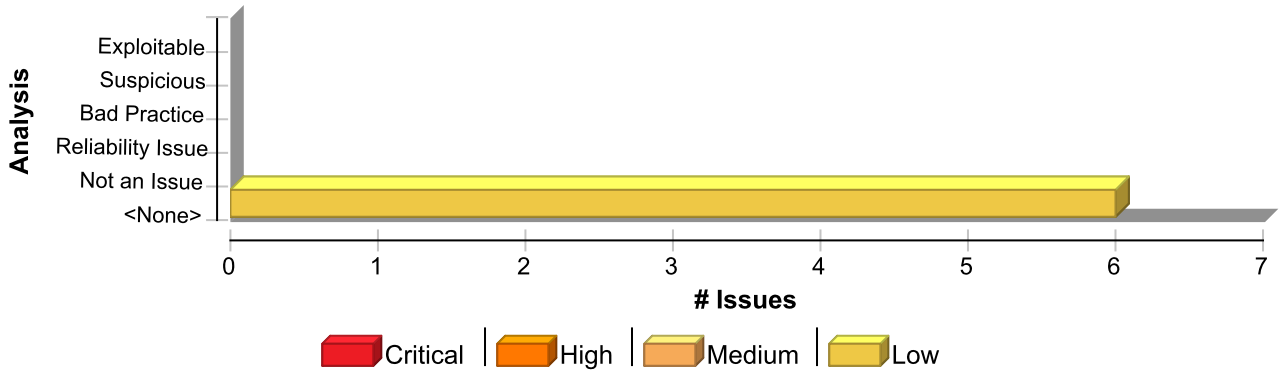
The `==` and `!=` operators will only behave as expected when they are used to compare strings contained in objects that are equal. The most common way for this to occur is for the strings to be interned, whereby the strings are added to a pool of objects maintained by the `String` class. Once a string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

Recommendation

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

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

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Code Correctness: Erroneous String Compare	6	0	0	6
Total	**6**	**0**	**0**	**6**

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.client

client/ClusterClient.scala, line 581 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: ClusterClientReceptionist()
File: client/ClusterClient.scala:581
Taint Flags:

```
578 final class ClusterClientReceptionist(system: ExtendedActorSystem) extends Extension {
579
580  private val config = system.settings.config.getConfig("akka.cluster.client.receptionist")
581  private val role: Option[String] = config.getString("role") match {
582  case "" => None
583  case r => Some(r)
584  }
```

client/ClusterClient.scala, line 73 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: client/ClusterClient.scala:73
Taint Flags:

```
70  heartbeatInterval = config.getDuration("heartbeat-interval", MILLISECONDS).millis,
71  acceptableHeartbeatPause = config.getDuration("acceptable-heartbeat-pause", MILLISECONDS).millis,
72  bufferSize = config.getInt("buffer-size"),
73  reconnectTimeout = config.getString("reconnect-timeout") match {
74  case "off" => None
75  case _ => Some(config.getDuration("reconnect-timeout", MILLISECONDS).millis)
76  })
```

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 50 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 50 (Code Correctness: Erroneous String Compare) — Low

Sink Details

Sink: Operation
Enclosing Method: apply()
File: pubsub/DistributedPubSubMediator.scala:50
Taint Flags:

```
47 def apply(config: Config): DistributedPubSubSettings =
48 new DistributedPubSubSettings(
49 role = roleOption(config.getString("role")),
50 routingLogic = config.getString("routing-logic") match {
51 case "random" => RandomRoutingLogic()
52 case "round-robin" => RoundRobinRoutingLogic()
53 case "consistent-hashing" =>
```

pubsub/DistributedPubSubMediator.scala, line 50 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: pubsub/DistributedPubSubMediator.scala:50
Taint Flags:

```
47 def apply(config: Config): DistributedPubSubSettings =
48 new DistributedPubSubSettings(
49 role = roleOption(config.getString("role")),
50 routingLogic = config.getString("routing-logic") match {
51 case "random" => RandomRoutingLogic()
52 case "round-robin" => RoundRobinRoutingLogic()
53 case "consistent-hashing" =>
```

pubsub/DistributedPubSubMediator.scala, line 50 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()

Code Correctness: Erroneous String Compare	Low

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 50 (Code Correctness: Erroneous String Compare) — Low

File: pubsub/DistributedPubSubMediator.scala:50
Taint Flags:

```
47 def apply(config: Config): DistributedPubSubSettings =
48 new DistributedPubSubSettings(
49 role = roleOption(config.getString("role")),
50 routingLogic = config.getString("routing-logic") match {
51 case "random" => RandomRoutingLogic()
52 case "round-robin" => RoundRobinRoutingLogic()
53 case "consistent-hashing" =>
```

pubsub/DistributedPubSubMediator.scala, line 50 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: pubsub/DistributedPubSubMediator.scala:50
Taint Flags:

```
47 def apply(config: Config): DistributedPubSubSettings =
48 new DistributedPubSubSettings(
49 role = roleOption(config.getString("role")),
50 routingLogic = config.getString("routing-logic") match {
51 case "random" => RandomRoutingLogic()
52 case "round-robin" => RoundRobinRoutingLogic()
53 case "consistent-hashing" =>
```

Code Correctness: Non-Static Inner Class Implements Serializable (17 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

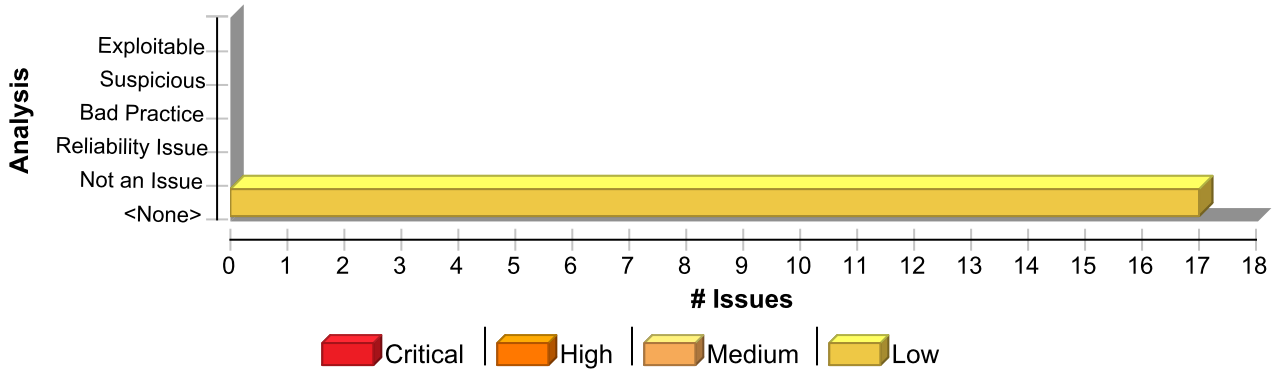
When using inner classes, they should not be serialized, or they should be changed to static-nested classes, since these do not have the drawbacks that non-static inner classes have when serialized. When a nested class is static it inherently has no association with instance variables (including those 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	17	0	0	17
Total	**17**	**0**	**0**	**17**

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

Package: akka.cluster.client

client/ClusterClient.scala, line 313 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterClient$SendToAll
File: client/ClusterClient.scala:313
Taint Flags:

```
310 @deprecated(
311 "Use Akka gRPC instead, see https://doc.akka.io/docs/akka/2.6/cluster-client.html#migration-to-akka-grpc",
312 since = "2.6.0")
313 final case class SendToAll(path: String, msg: Any)
314
315 @SerialVersionUID(1L)
316 @deprecated(
```

client/ClusterClient.scala, line 319 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterClient$Publish

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

Package: akka.cluster.client

client/ClusterClient.scala, line 319 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

File: client/ClusterClient.scala:319
Taint Flags:

```
316 @deprecated(
317 "Use Akka gRPC instead, see https://doc.akka.io/docs/akka/2.6/cluster-client.html#migration-to-akka-grpc",
318 since = "2.6.0")
319 final case class Publish(topic: String, msg: Any)
320
321 /**
322 * INTERNAL API
```

client/ClusterClient.scala, line 302 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterClient$Send
File: client/ClusterClient.scala:302
Taint Flags:

```
299 @deprecated(
300 "Use Akka gRPC instead, see https://doc.akka.io/docs/akka/2.6/cluster-client.html#migration-to-akka-grpc",
301 since = "2.6.0")
302 final case class Send(path: String, msg: Any, localAffinity: Boolean) {
303
304 /**
305 * Convenience constructor with `localAffinity` false
```

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 192 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$Publish
File: pubsub/DistributedPubSubMediator.scala:192
Taint Flags:

```
189 }
```

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

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 192 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
190 @SerialVersionUID(1L) final case class SubscribeAck(subscribe: Subscribe) extends DeadLetterSuppression
191 @SerialVersionUID(1L) final case class UnsubscribeAck(unsubscribe: Unsubscribe)
192 @SerialVersionUID(1L) final case class Publish(topic: String, msg: Any, sendOneMessageToEachGroup: Boolean)
193 extends DistributedPubSubMessage
194 with WrappedMessage {
195 def this(topic: String, msg: Any) = this(topic, msg, sendOneMessageToEachGroup = false)
```

pubsub/DistributedPubSubMediator.scala, line 199 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$Publish
File: pubsub/DistributedPubSubMediator.scala:199
Taint Flags:

```
196
197 override def message: Any = msg
198 }
199 object Publish {
200 def apply(topic: String, msg: Any) = new Publish(topic, msg)
201 }
202 @SerialVersionUID(1L) final case class Send(path: String, msg: Any, localAffinity: Boolean)
```

pubsub/DistributedPubSubMediator.scala, line 191 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$UnsubscribeAck
File: pubsub/DistributedPubSubMediator.scala:191
Taint Flags:

```
188 def apply(topic: String, ref: ActorRef) = new Unsubscribe(topic, ref)
189 }
190 @SerialVersionUID(1L) final case class SubscribeAck(subscribe: Subscribe) extends DeadLetterSuppression
191 @SerialVersionUID(1L) final case class UnsubscribeAck(unsubscribe: Unsubscribe)
192 @SerialVersionUID(1L) final case class Publish(topic: String, msg: Any, sendOneMessageToEachGroup: Boolean)
193 extends DistributedPubSubMessage
```

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

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 191 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
194 with WrappedMessage {
```

pubsub/DistributedPubSubMediator.scala, line 164 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$Put
File: pubsub/DistributedPubSubMediator.scala:164
Taint Flags:

```
161 def props(settings: DistributedPubSubSettings): Props =
162 Props(new DistributedPubSubMediator(settings)).withDeploy(Deploy.local)
163
164 @SerialVersionUID(1L) final case class Put(ref: ActorRef)
165 @SerialVersionUID(1L) final case class Remove(path: String)
166 @SerialVersionUID(1L) final case class Subscribe(topic: String, group: Option[String], ref: ActorRef) {
167 require(topic != null && topic != "", "topic must be defined")
```

pubsub/DistributedPubSubMediator.scala, line 166 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$Subscribe
File: pubsub/DistributedPubSubMediator.scala:166
Taint Flags:

```
163
164 @SerialVersionUID(1L) final case class Put(ref: ActorRef)
165 @SerialVersionUID(1L) final case class Remove(path: String)
166 @SerialVersionUID(1L) final case class Subscribe(topic: String, group: Option[String], ref: ActorRef) {
167 require(topic != null && topic != "", "topic must be defined")
168
169 /**
```

pubsub/DistributedPubSubMediator.scala, line 179 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

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

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 179 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$Subscribe
File: pubsub/DistributedPubSubMediator.scala:179
Taint Flags:

```
176 */
177 def this(topic: String, group: String, ref: ActorRef) = this(topic, Some(group), ref)
178 }
179 object Subscribe {
180 def apply(topic: String, ref: ActorRef) = new Subscribe(topic, ref)
181 }
182 @SerialVersionUID(1L) final case class Unsubscribe(topic: String, group: Option[String], ref: ActorRef) {
```

pubsub/DistributedPubSubMediator.scala, line 202 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$Send
File: pubsub/DistributedPubSubMediator.scala:202
Taint Flags:

```
199 object Publish {
200 def apply(topic: String, msg: Any) = new Publish(topic, msg)
201 }
202 @SerialVersionUID(1L) final case class Send(path: String, msg: Any, localAffinity: Boolean)
203 extends DistributedPubSubMessage
204 with WrappedMessage {
205 if (msg == null)
```

pubsub/DistributedPubSubMediator.scala, line 190 (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.pubsub

pubsub/DistributedPubSubMediator.scala, line 190 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: DistributedPubSubMediator$SubscribeAck
File: pubsub/DistributedPubSubMediator.scala:190
Taint Flags:

```
187 object Unsubscribe {
188 def apply(topic: String, ref: ActorRef) = new Unsubscribe(topic, ref)
189 }
190 @SerialVersionUID(1L) final case class SubscribeAck(subscribe: Subscribe) extends DeadLetterSuppression
191 @SerialVersionUID(1L) final case class UnsubscribeAck(unsubscribe: Unsubscribe)
192 @SerialVersionUID(1L) final case class Publish(topic: String, msg: Any, sendOneMessageToEachGroup: Boolean)
193 extends DistributedPubSubMessage
```

pubsub/DistributedPubSubMediator.scala, line 247 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$CurrentTopics
File: pubsub/DistributedPubSubMediator.scala:247
Taint Flags:

```
244 * Reply to `GetTopics`.
245 */
246 @SerialVersionUID(1L)
247 final case class CurrentTopics(topics: Set[String]) {
248
249 /**
250 * Java API
```

pubsub/DistributedPubSubMediator.scala, line 165 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$Remove
File: pubsub/DistributedPubSubMediator.scala:165
Taint Flags:

```
162 Props(new DistributedPubSubMediator(settings)).withDeploy(Deploy.local)
163
```

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

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 165 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
164 @SerialVersionUID(1L) final case class Put(ref: ActorRef)
165 @SerialVersionUID(1L) final case class Remove(path: String)
166 @SerialVersionUID(1L) final case class Subscribe(topic: String, group: Option[String], ref: ActorRef) {
167 require(topic != null && topic != "", "topic must be defined")
168
```

pubsub/DistributedPubSubMediator.scala, line 182 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$Unsubscribe
File: pubsub/DistributedPubSubMediator.scala:182
Taint Flags:

```
179 object Subscribe {
180 def apply(topic: String, ref: ActorRef) = new Subscribe(topic, ref)
181 }
182 @SerialVersionUID(1L) final case class Unsubscribe(topic: String, group: Option[String], ref: ActorRef) {
183 require(topic != null && topic != "", "topic must be defined")
184 def this(topic: String, ref: ActorRef) = this(topic, None, ref)
185 def this(topic: String, group: String, ref: ActorRef) = this(topic, Some(group), ref)
```

pubsub/DistributedPubSubMediator.scala, line 187 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$Unsubscribe
File: pubsub/DistributedPubSubMediator.scala:187
Taint Flags:

```
184 def this(topic: String, ref: ActorRef) = this(topic, None, ref)
185 def this(topic: String, group: String, ref: ActorRef) = this(topic, Some(group), ref)
186 }
187 object Unsubscribe {
188 def apply(topic: String, ref: ActorRef) = new Unsubscribe(topic, ref)
189 }
190 @SerialVersionUID(1L) final case class SubscribeAck(subscribe: Subscribe) extends DeadLetterSuppression
```

Code Correctness: Non-Static Inner Class Implements Serializable	Low

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 187 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low

pubsub/DistributedPubSubMediator.scala, line 215 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$SendToAll
File: pubsub/DistributedPubSubMediator.scala:215
Taint Flags:

```
212
213 override def message: Any = msg
214 }
215 @SerialVersionUID(1L) final case class SendToAll(path: String, msg: Any, allButSelf: Boolean = false)
216 extends DistributedPubSubMessage
217 with WrappedMessage {
218 if (msg == null)
```

pubsub/DistributedPubSubMediator.scala, line 273 (Code Correctness: Non-Static Inner Class Implements Serializable)	Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DistributedPubSubMediator$CountSubscribers
File: pubsub/DistributedPubSubMediator.scala:273
Taint Flags:

```
270 */
271 def getCountInstance: Count = Count
272
273 final case class CountSubscribers(topic: String)
274
275 /**
276 * INTERNAL API
```

Insecure Randomness (1 issue)

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.

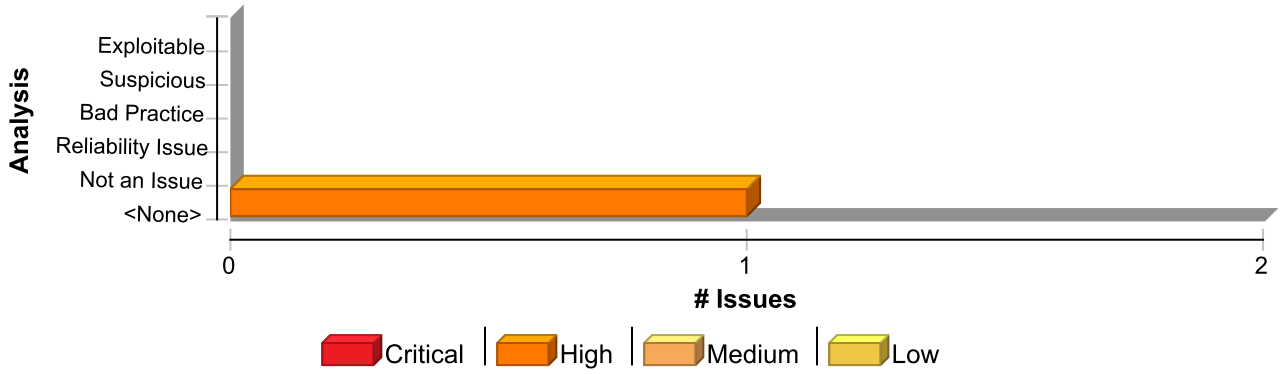
```
String GenerateReceiptURL(String baseUrl) {
    Random ranGen = new Random();
    ranGen.setSeed((new Date()).getTime());
    return (baseUrl + ranGen.nextInt(400000000) + ".html");
}
```

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	1	0	0	1
Total	**1**	**0**	**0**	**1**

Insecure Randomness — High

Package: akka.cluster.pubsub

pubsub/DistributedPubSubMediator.scala, line 905 (Insecure Randomness) — High

Issue Details

Kingdom: Security Features
Scan Engine: SCA (Semantic)

Sink Details

Sink: nextInt()
Enclosing Method: selectRandomNode()
File: pubsub/DistributedPubSubMediator.scala:905
Taint Flags:

```
902 }
903
904 def selectRandomNode(addresses: immutable.IndexedSeq[Address]): Option[Address] =
905 if (addresses.isEmpty) None else Some(addresses(ThreadLocalRandom.current.nextInt(addresses.size)))
906
907 def prune(): Unit = {
908 registry.foreach {
```