Fortify Standalone Report Generator

Developer Workbook

akka-cluster-sharding

Table of Contents

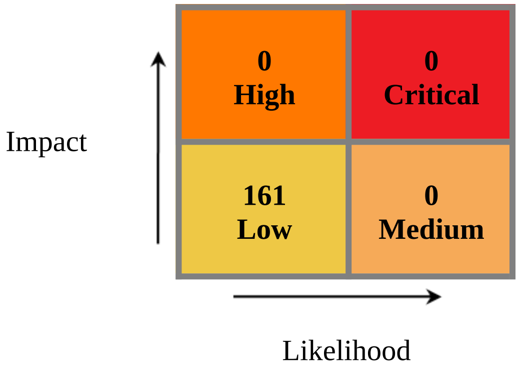
Executive Summary

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

Issues by Priority



0 **High**	**0** **Critical**
161 **Low**	**0** **Medium**

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, 11:21 AM	**Engine Version:**	21.1.1.0009
Host Name:	Jacks-Work-MBP.local	**Certification:**	VALID
Number of Files:	33	**Lines of Code:**	4,397

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 / 78	0 / 78
Code Correctness: Erroneous String Compare	0	0	0	0 / 24	0 / 24
Code Correctness: Non-Static Inner Class Implements Serializable	0	0	0	0 / 58	0 / 58
J2EE Bad Practices: Leftover Debug Code	0	0	0	0 / 1	0 / 1

Results Outline

Code Correctness: Constructor Invokes Overridable Function (78 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.

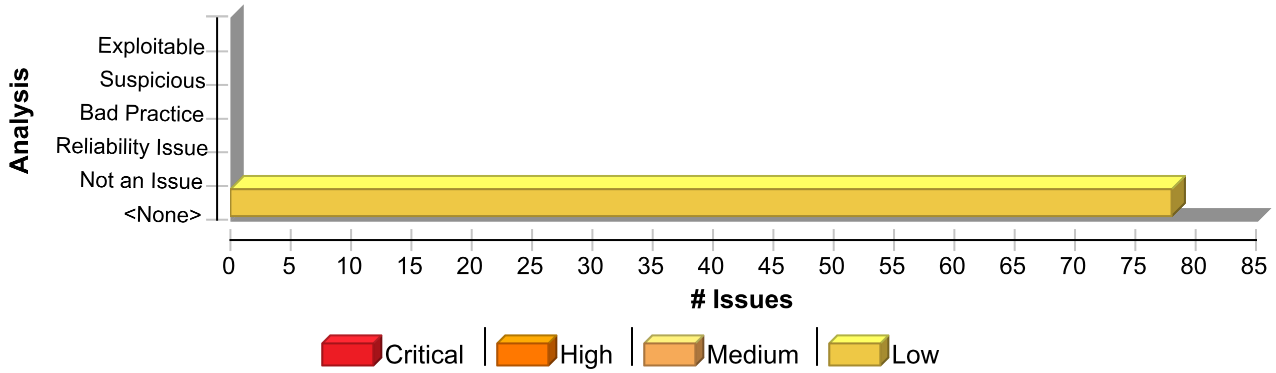
Recommendation

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	78	0	0	78
Total	**78**	**0**	**0**	**78**

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

Package: akka.cluster.sharding

Shard.scala, line 451 (Code Correctness: Constructor Invokes Overridable Function) **Low**

Issue Details

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding

Shard.scala, line 451 (Code Correctness: Constructor Invokes Overridable Function) — Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka$cluster$sharding$Shard$$verboseDebug
Enclosing Method: Shard()
File: Shard.scala:451
Taint Flags:

```
448  private val entities = {
449  val failOnInvalidStateTransition =
450  context.system.settings.config.getBoolean("akka.cluster.sharding.fail-on-invalid-entity-state-transition")
451  new Entities(log, settings.rememberEntities, verboseDebug, failOnInvalidStateTransition)
452  }
453
454  // Messages are buffered while an entity is passivating or waiting for a response
```

ShardCoordinator.scala, line 1496 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: getAllRememberedShards
Enclosing Method: DDataShardCoordinator()
File: ShardCoordinator.scala:1496
Taint Flags:

```
1493  // get state from ddata replicator, repeat until GetSuccess
1494  getCoordinatorState()
1495  if (settings.rememberEntities)
1496  getAllRememberedShards()
1497
1498  override def receive: Receive =
1499  waitingForInitialState(Set.empty)
```

ShardCoordinator.scala, line 1474 (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.sharding

ShardCoordinator.scala, line 1474 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink: FunctionCall: node
Enclosing Method: DDataShardCoordinator()
File: ShardCoordinator.scala:1474
Taint Flags:

```
1471  }
1472
1473  implicit val node: Cluster = Cluster(context.system)
1474  private implicit val selfUniqueAddress: SelfUniqueAddress = SelfUniqueAddress(node.selfUniqueAddress)
1475  private val CoordinatorStateKey = LWWRegisterKey[State](s"${typeName}CoordinatorState")
1476  private val initEmptyState = State.empty.withRememberEntities(settings.rememberEntities)
1477
```

ShardCoordinator.scala, line 1491 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node
Enclosing Method: DDataShardCoordinator()
File: ShardCoordinator.scala:1491
Taint Flags:

```
1488  RememberEntitiesStoreStopped)
1489  }
1490  private val rememberEntities = rememberEntitiesStore.isDefined
1491  node.subscribe(self, ClusterEvent.InitialStateAsEvents, ClusterShuttingDown.getClass)
1492
1493  // get state from ddata replicator, repeat until GetSuccess
1494  getCoordinatorState()
```

ShardRegion.scala, line 651 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: initRegistrationDelay
Enclosing Method: ShardRegion()
File: ShardRegion.scala:651
Taint Flags:

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding

ShardRegion.scala, line 651 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
648 import context.dispatcher
649 var retryCount = 0
650 val initRegistrationDelay: FiniteDuration = 100.millis.max(retryInterval / 2 / 2 / 2)
651 var nextRegistrationDelay: FiniteDuration = initRegistrationDelay
652
653 // for CoordinatedShutdown
654 val gracefulShutdownProgress = Promise[Done]()
```

ShardRegion.scala, line 701 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: cluster
Enclosing Method: ShardRegion()
File: ShardRegion.scala:701
Taint Flags:

```
698 // when using proxy the data center can be different from the own data center
699 private val targetDcRole = dataCenter match {
700 case Some(t) => ClusterSettings.DcRolePrefix + t
701 case None => ClusterSettings.DcRolePrefix + cluster.settings.SelfDataCenter
702 }
703
704 def matchingRole(member: Member): Boolean =
```

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

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ageOrdering
Enclosing Method: ShardRegion()
File: ShardRegion.scala:632
Taint Flags:

```
629
630 // sort by age, oldest first
631 val ageOrdering = Member.ageOrdering
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding

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

```
632  var membersByAge: immutable.SortedSet[Member] = immutable.SortedSet.empty(ageOrdering)
633  // membersByAge contains members with these status
634  private val memberStatusOfInterest: Set[MemberStatus] =
635  Set(MemberStatus.Up, MemberStatus.Leaving, MemberStatus.Exiting)
```

ClusterShardingSettings.scala, line 1259 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: StateStoreModePersistence
Enclosing Method: ClusterShardingSettings()
File: ClusterShardingSettings.scala:1259
Taint Flags:

```
1256
1257  import ClusterShardingSettings.{ RememberEntitiesStoreCustom, StateStoreModeDData, StateStoreModePersistence }
1258  require(
1259  stateStoreMode == StateStoreModePersistence || stateStoreMode == StateStoreModeDData || stateStoreMode == RememberEntitiesStoreCustom,
1260  s"Unknown 'state-store-mode' [$stateStoreMode], valid values are '$StateStoreModeDData' or '$StateStoreModePersistence'")
1261
1262  /** If true, this node should run the shard region, otherwise just a shard proxy should started on this node. */
```

ShardCoordinator.scala, line 1490 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: rememberEntitiesStore
Enclosing Method: DDataShardCoordinator()
File: ShardCoordinator.scala:1490
Taint Flags:

```
1487  context.actorOf(provider.coordinatorStoreProps(), "RememberEntitiesStore"),
1488  RememberEntitiesStoreStopped)
1489  }
1490  private val rememberEntities = rememberEntitiesStore.isDefined
1491  node.subscribe(self, ClusterEvent.InitialStateAsEvents, ClusterShuttingDown.getClass)
1492
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding

ShardCoordinator.scala, line 1490 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
1493 // get state from ddata replicator, repeat until GetSuccess
```

ClusterShardingSettings.scala, line 1259 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: RememberEntitiesStoreCustom
Enclosing Method: ClusterShardingSettings()
File: ClusterShardingSettings.scala:1259
Taint Flags:

```
1256
1257 import ClusterShardingSettings.{ RememberEntitiesStoreCustom, StateStoreModeDData, StateStoreModePersistence }
1258 require(
1259 stateStoreMode == StateStoreModePersistence || stateStoreMode == StateStoreModeDData || stateStoreMode ==
RememberEntitiesStoreCustom,
1260 s"Unknown 'state-store-mode' [$stateStoreMode], valid values are '$StateStoreModeDData' or '$StateStoreModePersistence'")
1261
1262 /** If true, this node should run the shard region, otherwise just a shard proxy should started on this node. */
```

ClusterShardingSettings.scala, line 1259 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: StateStoreModeDData
Enclosing Method: ClusterShardingSettings()
File: ClusterShardingSettings.scala:1259
Taint Flags:

```
1256
1257 import ClusterShardingSettings.{ RememberEntitiesStoreCustom, StateStoreModeDData, StateStoreModePersistence }
1258 require(
1259 stateStoreMode == StateStoreModePersistence || stateStoreMode == StateStoreModeDData || stateStoreMode ==
RememberEntitiesStoreCustom,
1260 s"Unknown 'state-store-mode' [$stateStoreMode], valid values are '$StateStoreModeDData' or '$StateStoreModePersistence'")
1261
1262 /** If true, this node should run the shard region, otherwise just a shard proxy should started on this node. */
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding

ShardCoordinator.scala, line 1494 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: getCoordinatorState
Enclosing Method: DDataShardCoordinator()
File: ShardCoordinator.scala:1494
Taint Flags:

```
1491  node.subscribe(self, ClusterEvent.InitialStateAsEvents, ClusterShuttingDown.getClass)
1492
1493  // get state from ddata replicator, repeat until GetSuccess
1494  getCoordinatorState()
1495  if (settings.rememberEntities)
1496  getAllRememberedShards()
1497
```

Shard.scala, line 464 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: passivationStrategy
Enclosing Method: Shard()
File: Shard.scala:464
Taint Flags:

```
461  private val passivationStrategy = EntityPassivationStrategy(settings)
462
463  import context.dispatcher
464  private val passivateIntervalTask = passivationStrategy.scheduledInterval.map { interval =>
465  context.system.scheduler.scheduleWithFixedDelay(interval, interval, self, PassivateIntervalTick)
466  }
467
```

Package: akka.cluster.sharding.external

external/ExternalShardAllocationStrategy.scala, line 98 (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.sharding.external

external/ExternalShardAllocationStrategy.scala, line 98 (Code Correctness: Constructor Invokes Overridable Function)

Low

Sink Details

Sink: FunctionCall: system
Enclosing Method: ExternalShardAllocationStrategy()
File: external/ExternalShardAllocationStrategy.scala:98
Taint Flags:

```
95
96  import akka.pattern.ask
97
98  private val log = Logging(system, classOf[ExternalShardAllocationStrategy])
99
100  private var shardState: ActorRef = _
101
```

external/ExternalShardAllocationStrategy.scala, line 108 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: system
Enclosing Method: ExternalShardAllocationStrategy()
File: external/ExternalShardAllocationStrategy.scala:108
Taint Flags:

```
105  .systemActorOf(DDataStateActor.props(typeName), s"external-allocation-state-$typeName")
106  }
107
108  private val cluster = Cluster(system)
109
110  override def start(): Unit = {
111  shardState = createShardStateActor()
```

Package: akka.cluster.sharding.external.internal

external/internal/ExternalShardAllocationClientImpl.scala, line 60 (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.sharding.external.internal

external/internal/ExternalShardAllocationClientImpl.scala, line 60 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink: FunctionCall: timeout
Enclosing Method: ExternalShardAllocationClientImpl()
File: external/internal/ExternalShardAllocationClientImpl.scala:60
Taint Flags:

```
57 system.settings.config
58 .getDuration("akka.cluster.sharding.external-shard-allocation-strategy.client-timeout")
59 .asScala
60 private implicit val askTimeout: Timeout = Timeout(timeout * 2)
61 private implicit val ec: MessageDispatcher = system.dispatchers.internalDispatcher
62
63 private val Key = ExternalShardAllocationStrategy.ddataKey(typeName)
```

Package: akka.cluster.sharding.internal

internal/EntityPassivationStrategy.scala, line 663 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: calculateLimits
Enclosing Method: CompositeEntityPassivationStrategy()
File: internal/EntityPassivationStrategy.scala:663
Taint Flags:

```
660 }
661
662 // set initial limits based on initial window proportion
663 calculateLimits()
664 window.updateLimit(windowLimit)
665 main.updateLimit(mainLimit)
666
```

internal/DDataRememberEntitiesCoordinatorStore.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node
Enclosing Method: DDataRememberEntitiesCoordinatorStore()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.internal

internal/DDataRememberEntitiesCoordinatorStore.scala, line 43 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: internal/DDataRememberEntitiesCoordinatorStore.scala:43
Taint Flags:

```
40 with ActorLogging {
41
42 implicit val node: Cluster = Cluster(context.system)
43 implicit val selfUniqueAddress: SelfUniqueAddress = SelfUniqueAddress(node.selfUniqueAddress)
44
45 private val readMajority = ReadMajority(settings.tuningParameters.waitingForStateTimeout, majorityMinCap)
46 private val writeMajority = WriteMajority(settings.tuningParameters.updatingStateTimeout, majorityMinCap)
```

internal/EntityPassivationStrategy.scala, line 488 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: segmentLimits
Enclosing Method: SegmentedLeastRecentlyUsedReplacementPolicy()
File: internal/EntityPassivationStrategy.scala:488
Taint Flags:

```
485 }
486
487 private val segmentedRecencyList =
488 if (idleEnabled) SegmentedRecencyList.withOverallRecency.empty[EntityId](segmentLimits)
489 else SegmentedRecencyList.empty[EntityId](segmentLimits)
490
491 override def size: Int = segmentedRecencyList.size
```

internal/EntityPassivationStrategy.scala, line 489 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: segmentLimits
Enclosing Method: SegmentedLeastRecentlyUsedReplacementPolicy()
File: internal/EntityPassivationStrategy.scala:489
Taint Flags:

```
486
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.internal

internal/EntityPassivationStrategy.scala, line 489 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
487  private val segmentedRecencyList =
488  if (idleEnabled) SegmentedRecencyList.withOverallRecency.empty[EntityId](segmentLimits)
489  else SegmentedRecencyList.empty[EntityId](segmentLimits)
490
491  override def size: Int = segmentedRecencyList.size
492
```

internal/EntityPassivationStrategy.scala, line 665 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: mainLimit
Enclosing Method: CompositeEntityPassivationStrategy()
File: internal/EntityPassivationStrategy.scala:665
Taint Flags:

```
662  // set initial limits based on initial window proportion
663  calculateLimits()
664  window.updateLimit(windowLimit)
665  main.updateLimit(mainLimit)
666
667  override def entityTouched(id: EntityId): PassivateEntities = {
668  admissionFilter.update(id)
```

internal/RememberEntityStarter.scala, line 74 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: idsLeftToStart_=
Enclosing Method: RememberEntityStarter()
File: internal/RememberEntityStarter.scala:74
Taint Flags:

```
71  startBatch(ids)
72  case "constant" =>
73  import settings.tuningParameters
74  idsLeftToStart = ids
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.internal

internal/RememberEntityStarter.scala, line 74 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
75 timers.startTimerWithFixedDelay(
76 "constant",
77 StartBatch(tuningParameters.entityRecoveryConstantRateStrategyNumberOfEntities),
```

internal/RememberEntityStarter.scala, line 70 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: idsLeftToStart_=
Enclosing Method: RememberEntityStarter()
File: internal/RememberEntityStarter.scala:70
Taint Flags:

```
67
68 settings.tuningParameters.entityRecoveryStrategy match {
69 case "all" =>
70 idsLeftToStart = Set.empty
71 startBatch(ids)
72 case "constant" =>
73 import settings.tuningParameters
```

internal/DDataRememberEntitiesCoordinatorStore.scala, line 56 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: getAllShards
Enclosing Method: DDataRememberEntitiesCoordinatorStore()
File: internal/DDataRememberEntitiesCoordinatorStore.scala:56
Taint Flags:

```
53 def getAllShards(): Unit = {
54 replicator ! Replicator.Get(AllShardsKey, readMajority)
55 }
56 getAllShards()
57
58 override def receive: Receive = {
59 case RememberEntitiesCoordinatorStore.GetShards =>
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.internal

internal/CustomStateStoreModeProvider.scala, line 27 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: log
Enclosing Method: CustomStateStoreModeProvider()
File: internal/CustomStateStoreModeProvider.scala:27
Taint Flags:

```
24 extends RememberEntitiesProvider {
25
26 private val log = Logging(system, classOf[CustomStateStoreModeProvider])
27 log.warning("Using custom remember entities store for [{}], not intended for production use.", typeName)
28 val customStore = if (system.settings.config.hasPath("akka.cluster.sharding.remember-entities-custom-store")) {
29 val customClassName = system.settings.config.getString("akka.cluster.sharding.remember-entities-custom-store")
30
```

internal/CustomStateStoreModeProvider.scala, line 37 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: log
Enclosing Method: CustomStateStoreModeProvider()
File: internal/CustomStateStoreModeProvider.scala:37
Taint Flags:

```
34 .createInstanceFor[RememberEntitiesProvider](
35 customClassName,
36 Vector((classOf[ClusterShardingSettings], settings), (classOf[String], typeName)))
37 log.debug("Will use custom remember entities store provider [{}]", store)
38 store.get
39
40 } else {
```

internal/CustomStateStoreModeProvider.scala, line 41 (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.sharding.internal

internal/CustomStateStoreModeProvider.scala, line 41 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: log
Enclosing Method: CustomStateStoreModeProvider()
File: internal/CustomStateStoreModeProvider.scala:41
Taint Flags:

```
38 store.get
39
40 } else {
41 log.error("Missing custom store class configuration for CustomStateStoreModeProvider")
42 throw new RuntimeException("Missing custom store class configuration")
43 }
44
```

internal/RememberEntityStarter.scala, line 79 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka$cluster$sharding$internal$RememberEntityStarter$$startBatch
Enclosing Method: RememberEntityStarter()
File: internal/RememberEntityStarter.scala:79
Taint Flags:

```
76 "constant",
77 StartBatch(tuningParameters.entityRecoveryConstantRateStrategyNumberOfEntities),
78 tuningParameters.entityRecoveryConstantRateStrategyFrequency)
79 startBatch(tuningParameters.entityRecoveryConstantRateStrategyNumberOfEntities)
80 }
81 timers.startTimerWithFixedDelay("retry", ResendUnAcked, settings.tuningParameters.retryInterval)
82
```

internal/RememberEntityStarter.scala, line 71 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: startBatch
Enclosing Method: RememberEntityStarter()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.internal

internal/RememberEntityStarter.scala, line 71 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: internal/RememberEntityStarter.scala:71
Taint Flags:

```
68 settings.tuningParameters.entityRecoveryStrategy match {
69 case "all" =>
70 idsLeftToStart = Set.empty
71 startBatch(ids)
72 case "constant" =>
73 import settings.tuningParameters
74 idsLeftToStart = ids
```

internal/EntityPassivationStrategy.scala, line 968 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: createSketch
Enclosing Method: FastFrequencySketchAdmissionFilter()
File: internal/EntityPassivationStrategy.scala:968
Taint Flags:

```
965 private def createSketch(capacity: Int): FastFrequencySketch[EntityId] =
966 FastFrequencySketch[EntityId](capacity, widthMultiplier, resetMultiplier)
967
968 private var frequencySketch = createSketch(initialCapacity)
969
970 override def updateCapacity(newCapacity: Int): Unit = frequencySketch = createSketch(newCapacity)
971
```

internal/EntityPassivationStrategy.scala, line 664 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: windowLimit
Enclosing Method: CompositeEntityPassivationStrategy()
File: internal/EntityPassivationStrategy.scala:664
Taint Flags:

```
661
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.internal

internal/EntityPassivationStrategy.scala, line 664 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
662  // set initial limits based on initial window proportion
663  calculateLimits()
664  window.updateLimit(windowLimit)
665  main.updateLimit(mainLimit)
666
667  override def entityTouched(id: EntityId): PassivateEntities = {
```

internal/DDataRememberEntitiesShardStore.scala, line 94 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: akka$cluster$sharding$internal$DDataRememberEntitiesShardStore$$stateKeys
Enclosing Method: DDataRememberEntitiesShardStore()
File: internal/DDataRememberEntitiesShardStore.scala:94
Taint Flags:

```
91  // Note that the timeout is actually updatingStateTimeout / 4 so that we fit 3 retries and a response in the timeout before the shard sees it as a failure
92  private val writeMajority = WriteMajority(settings.tuningParameters.updatingStateTimeout / 4, majorityMinCap)
93  private val maxUpdateAttempts = 3
94  private val keys = stateKeys(typeName, shardId)
95
96  if (log.isDebugEnabled) {
97  log.debug(
```

internal/EntityPassivationStrategy.scala, line 943 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: createSketch
Enclosing Method: FrequencySketchAdmissionFilter()
File: internal/EntityPassivationStrategy.scala:943
Taint Flags:

```
940  private def createSketch(capacity: Int): FrequencySketch[EntityId] =
941  FrequencySketch[EntityId](capacity, widthMultiplier, resetMultiplier, depth, counterBits)
942
943  private var frequencySketch = createSketch(initialCapacity)
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.internal

internal/EntityPassivationStrategy.scala, line 943 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
944
945  override def updateCapacity(newCapacity: Int): Unit = frequencySketch = createSketch(newCapacity)
946
```

internal/DDataRememberEntitiesShardStore.scala, line 103 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: loadAllEntities
Enclosing Method: DDataRememberEntitiesShardStore()
File: internal/DDataRememberEntitiesShardStore.scala:103
Taint Flags:

```
100  settings.tuningParameters.updatingStateTimeout.pretty,
101  majorityMinCap)
102  }
103  loadAllEntities()
104
105  private def key(entityId: EntityId): ORSetKey[EntityId] = {
106  val i = math.abs(entityId.hashCode % numberOfKeys)
```

internal/DDataRememberEntitiesShardStore.scala, line 88 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: node
Enclosing Method: DDataRememberEntitiesShardStore()
File: internal/DDataRememberEntitiesShardStore.scala:88
Taint Flags:

```
85
86  implicit val ec: ExecutionContext = context.dispatcher
87  implicit val node: Cluster = Cluster(context.system)
88  implicit val selfUniqueAddress: SelfUniqueAddress = SelfUniqueAddress(node.selfUniqueAddress)
89
90  private val readMajority = ReadMajority(settings.tuningParameters.waitingForStateTimeout, majorityMinCap)
91  // Note that the timeout is actually updatingStateTimeout / 4 so that we fit 3 retries and a response in the timeout before the shard sees it as a failure
```

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.sharding.internal

internal/DDataRememberEntitiesShardStore.scala, line 88 (Code Correctness: Constructor Invokes Overridable Function)

Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardRegionProxyTerminatedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: RegionStoppedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: EntitiesStartedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetShardRegionStatsManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (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.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)

Low

Sink Details

Sink: FunctionCall: CurrentShardRegionStateManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardStatsManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetShardRegionStateManifest
Enclosing Method: ClusterShardingMessageSerializer()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardRegionRegisteredManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardHomeAllocatedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
```

Code Correctness: Constructor Invokes Overridable Function

Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)

Low

```
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: StartEntityManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: HandOffManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: RegisterAckManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: BeginHandOffManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: CoordinatorStateManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardRegionTerminatedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (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.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: EntitiesStoppedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: CurrentShardStateManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetCurrentRegionsManifest
Enclosing Method: ClusterShardingMessageSerializer()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetShardStatsManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetClusterShardingStatsManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
```

Code Correctness: Constructor Invokes Overridable Function | Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) | Low

```
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardStateManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: RegisterManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardRegionProxyRegisteredManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: EventSourcedRememberShardsMigrationMarkerManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardHomesManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: HostShardManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (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.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: ShardRegionStatsManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: EntityStartedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardHomeDeallocatedManifest
Enclosing Method: ClusterShardingMessageSerializer()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: StartEntityAckManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetCurrentShardStateManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GetShardHomeManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: RegisterProxyManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardHomeManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ShardStoppedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: GracefulShutdownReqManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: EntityStateManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (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.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Sink Details

Sink: FunctionCall: ShardStartedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: ClusterShardingStatsManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: EventSourcedRememberShardsState
Enclosing Method: ClusterShardingMessageSerializer()

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: BeginHandOffAckManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: CurrentRegionsManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
```

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.protobuf

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

```
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

protobuf/ClusterShardingMessageSerializer.scala, line 100 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: EntityStoppedManifest
Enclosing Method: ClusterShardingMessageSerializer()
File: protobuf/ClusterShardingMessageSerializer.scala:100
Taint Flags:

```
97  private val EventSourcedRememberShardsMigrationMarkerManifest = "SM"
98  private val EventSourcedRememberShardsState = "SS"
99
100  private val fromBinaryMap = collection.immutable.HashMap[String, Array[Byte] => AnyRef](
101  EntityStateManifest -> entityStateFromBinary,
102  EntityStartedManifest -> entityStartedFromBinary,
103  EntitiesStartedManifest -> entitiesStartedFromBinary,
```

Code Correctness: Erroneous String Compare (24 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");
  }
```

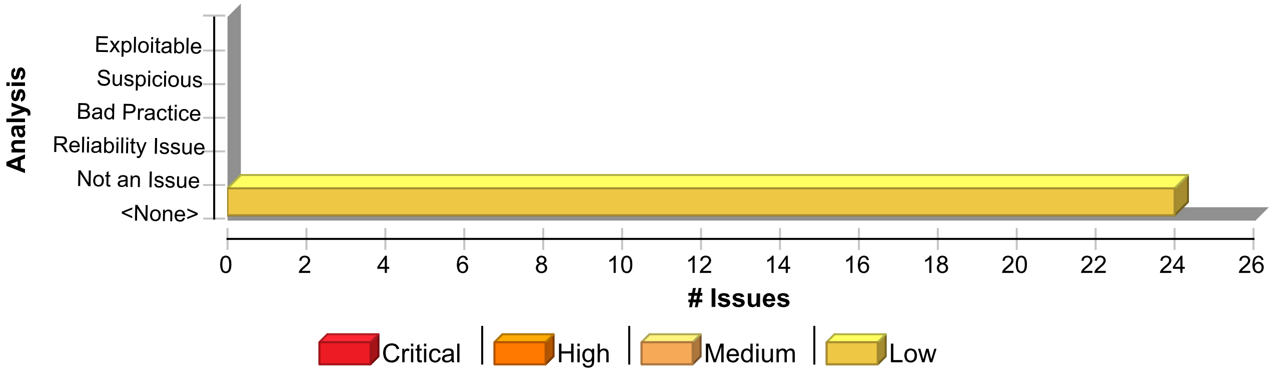
The `==` and `!=` operators will only behave as expected when they are used to compare strings contained in objects that are equal. The most common way for this to occur is for the strings to be interned, whereby the strings are added to a pool of objects maintained by the `String` class. Once a string 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	24	0	0	24
Total	**24**	**0**	**0**	**24**

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 253 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:253
Taint Flags:

```
250 }
251
252 def optional(config: Config): Option[IdleSettings] =
253 toRootLowerCase(config.getString("timeout")) match {
254 case "off" | "none" => None
255 case _ => Some(IdleSettings(config))
256 }
```

ClusterShardingSettings.scala, line 306 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:306
Taint Flags:

```
303 }
304
305 def optional(config: Config): Option[SegmentedSettings] = {
306 toRootLowerCase(config.getString("levels")) match {
307 case "off" | "none" => None
308 case _ => Some(SegmentedSettings(config))
309 }
```

ClusterShardingSettings.scala, line 275 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Code Correctness: Erroneous String Compare | Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 275 (Code Correctness: Erroneous String Compare) | Low

Sink: Operation
Enclosing Method: apply()
File: ClusterShardingSettings.scala:275
Taint Flags:

```
272
273 object PolicySettings {
274 def apply(config: Config): PolicySettings =
275 toRootLowerCase(config.getString("policy")) match {
276 case "least-recently-used" => LeastRecentlyUsedSettings(config.getConfig("least-recently-used"))
277 case "most-recently-used" => MostRecentlyUsedSettings(config.getConfig("most-recently-used"))
278 case "least-frequently-used" => LeastFrequentlyUsedSettings(config.getConfig("least-frequently-used"))
```

ClusterShardingSettings.scala, line 569 (Code Correctness: Erroneous String Compare) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: ClusterShardingSettings.scala:569
Taint Flags:

```
566 */
567 @ApiMayChange
568 def apply(config: Config): PassivationStrategySettings = {
569 toRootLowerCase(config.getString("strategy")) match {
570 case "off" | "none" => PassivationStrategySettings.disabled
571 case strategyName =>
572 val strategyDefaults = config.getConfig("strategy-defaults")
```

ClusterShardingSettings.scala, line 275 (Code Correctness: Erroneous String Compare) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: ClusterShardingSettings.scala:275
Taint Flags:

```
272
273 object PolicySettings {
```

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 275 (Code Correctness: Erroneous String Compare) — Low

```
274 def apply(config: Config): PolicySettings =
275 toRootLowerCase(config.getString("policy")) match {
276 case "least-recently-used" => LeastRecentlyUsedSettings(config.getConfig("least-recently-used"))
277 case "most-recently-used" => MostRecentlyUsedSettings(config.getConfig("most-recently-used"))
278 case "least-frequently-used" => LeastFrequentlyUsedSettings(config.getConfig("least-frequently-used"))
```

ClusterShardingSettings.scala, line 379 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:379
Taint Flags:

```
376
377 object FilterSettings {
378 def optional(config: Config): Option[FilterSettings] =
379 toRootLowerCase(config.getString("filter")) match {
380 case "off" | "none" => None
381 case "frequency-sketch" => Some(FrequencySketchSettings(config.getConfig("frequency-sketch")))
382 case _ => None
```

ClusterShardingSettings.scala, line 447 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:447
Taint Flags:

```
444 }
445
446 def optional(config: Config): Option[WindowSettings] =
447 toRootLowerCase(config.getString("policy")) match {
448 case "off" | "none" => None
449 case _ => Some(WindowSettings(config))
450 }
```

Code Correctness: Erroneous String Compare | Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 306 (Code Correctness: Erroneous String Compare) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:306
Taint Flags:

```
303 }
304
305 def optional(config: Config): Option[SegmentedSettings] = {
306 toRootLowerCase(config.getString("levels")) match {
307 case "off" | "none" => None
308 case _ => Some(SegmentedSettings(config))
309 }
```

ClusterShardingSettings.scala, line 486 (Code Correctness: Erroneous String Compare) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:486
Taint Flags:

```
483
484 object OptimizerSettings {
485 def optional(config: Config): Option[OptimizerSettings] =
486 toRootLowerCase(config.getString("optimizer")) match {
487 case "off" | "none" => None
488 case "hill-climbing" => Some(HillClimbingSettings(config.getConfig("hill-climbing")))
489 case _ => None
```

ClusterShardingSettings.scala, line 59 (Code Correctness: Erroneous String Compare) | Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 59 (Code Correctness: Erroneous String Compare) — Low

Sink: Operation
Enclosing Method: configMajorityPlus()
File: ClusterShardingSettings.scala:59
Taint Flags:

```
56  def apply(config: Config): ClusterShardingSettings = {
57
58  def configMajorityPlus(p: String): Int = {
59  toRootLowerCase(config.getString(p)) match {
60  case "all" => Int.MaxValue
61  case _ => config.getInt(p)
62  }
```

ClusterShardingSettings.scala, line 253 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:253
Taint Flags:

```
250  }
251
252  def optional(config: Config): Option[IdleSettings] =
253  toRootLowerCase(config.getString("timeout")) match {
254  case "off" | "none" => None
255  case _ => Some(IdleSettings(config))
256  }
```

ClusterShardingSettings.scala, line 575 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: ClusterShardingSettings.scala:575
Taint Flags:

```
572  val strategyDefaults = config.getConfig("strategy-defaults")
573  val strategyConfig = config.getConfig(strategyName).withFallback(strategyDefaults)
```

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 575 (Code Correctness: Erroneous String Compare) — Low

```
574 val idleEntitySettings = IdleSettings.optional(strategyConfig.getConfig("idle-entity"))
575 val activeEntityLimit = strategyConfig.getString("active-entity-limit") match {
576 case "off" | "none" => None
577 case _ => Some(strategyConfig.getInt("active-entity-limit"))
578 }
```

ClusterShardingSettings.scala, line 275 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: ClusterShardingSettings.scala:275
Taint Flags:

```
272
273 object PolicySettings {
274 def apply(config: Config): PolicySettings =
275 toRootLowerCase(config.getString("policy")) match {
276 case "least-recently-used" => LeastRecentlyUsedSettings(config.getConfig("least-recently-used"))
277 case "most-recently-used" => MostRecentlyUsedSettings(config.getConfig("most-recently-used"))
278 case "least-frequently-used" => LeastFrequentlyUsedSettings(config.getConfig("least-frequently-used"))
```

ClusterShardingSettings.scala, line 282 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:282
Taint Flags:

```
279 }
280
281 def optional(config: Config): Option[PolicySettings] =
282 toRootLowerCase(config.getString("policy")) match {
283 case "off" | "none" => None
284 case _ => Some(PolicySettings(config))
285 }
```

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 282 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:282
Taint Flags:

```
279 }
280
281 def optional(config: Config): Option[PolicySettings] =
282 toRootLowerCase(config.getString("policy")) match {
283 case "off" | "none" => None
284 case _ => Some(PolicySettings(config))
285 }
```

ClusterShardingSettings.scala, line 569 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: ClusterShardingSettings.scala:569
Taint Flags:

```
566 */
567 @ApiMayChange
568 def apply(config: Config): PassivationStrategySettings = {
569 toRootLowerCase(config.getString("strategy")) match {
570 case "off" | "none" => PassivationStrategySettings.disabled
571 case strategyName =>
572 val strategyDefaults = config.getConfig("strategy-defaults")
```

ClusterShardingSettings.scala, line 379 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 379 (Code Correctness: Erroneous String Compare) — Low

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:379
Taint Flags:

```
376
377  object FilterSettings {
378  def optional(config: Config): Option[FilterSettings] =
379  toRootLowerCase(config.getString("filter")) match {
380  case "off" | "none" => None
381  case "frequency-sketch" => Some(FrequencySketchSettings(config.getConfig("frequency-sketch")))
382  case _ => None
```

ClusterShardingSettings.scala, line 447 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:447
Taint Flags:

```
444  }
445
446  def optional(config: Config): Option[WindowSettings] =
447  toRootLowerCase(config.getString("policy")) match {
448  case "off" | "none" => None
449  case _ => Some(WindowSettings(config))
450  }
```

ClusterShardingSettings.scala, line 486 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:486
Taint Flags:

```
483
484  object OptimizerSettings {
```

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 486 (Code Correctness: Erroneous String Compare) — Low

```
485 def optional(config: Config): Option[OptimizerSettings] =
486 toRootLowerCase(config.getString("optimizer")) match {
487 case "off" | "none" => None
488 case "hill-climbing" => Some(HillClimbingSettings(config.getConfig("hill-climbing")))
489 case _ => None
```

ClusterShardingSettings.scala, line 575 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: apply()
File: ClusterShardingSettings.scala:575
Taint Flags:

```
572 val strategyDefaults = config.getConfig("strategy-defaults")
573 val strategyConfig = config.getConfig(strategyName).withFallback(strategyDefaults)
574 val idleEntitySettings = IdleSettings.optional(strategyConfig.getConfig("idle-entity"))
575 val activeEntityLimit = strategyConfig.getString("active-entity-limit") match {
576 case "off" | "none" => None
577 case _ => Some(strategyConfig.getInt("active-entity-limit"))
578 }
```

ClusterShardingSettings.scala, line 486 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:486
Taint Flags:

```
483
484 object OptimizerSettings {
485 def optional(config: Config): Option[OptimizerSettings] =
486 toRootLowerCase(config.getString("optimizer")) match {
487 case "off" | "none" => None
488 case "hill-climbing" => Some(HillClimbingSettings(config.getConfig("hill-climbing")))
489 case _ => None
```

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 379 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: optional()
File: ClusterShardingSettings.scala:379
Taint Flags:

```
376
377 object FilterSettings {
378 def optional(config: Config): Option[FilterSettings] =
379 toRootLowerCase(config.getString("filter")) match {
380 case "off" | "none" => None
381 case "frequency-sketch" => Some(FrequencySketchSettings(config.getConfig("frequency-sketch")))
382 case _ => None
```

Package: akka.cluster.sharding.internal

internal/RememberEntityStarter.scala, line 68 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Operation
Enclosing Method: RememberEntityStarter()
File: internal/RememberEntityStarter.scala:68
Taint Flags:

```
65 ids.size,
66 settings.tuningParameters.entityRecoveryStrategy)
67
68 settings.tuningParameters.entityRecoveryStrategy match {
69 case "all" =>
70 idsLeftToStart = Set.empty
71 startBatch(ids)
```

internal/RememberEntityStarter.scala, line 68 (Code Correctness: Erroneous String Compare) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Code Correctness: Erroneous String Compare — Low

Package: akka.cluster.sharding.internal

internal/RememberEntityStarter.scala, line 68 (Code Correctness: Erroneous String Compare) — Low

Sink Details

Sink: Operation
Enclosing Method: RememberEntityStarter()
File: internal/RememberEntityStarter.scala:68
Taint Flags:

```
65 ids.size,
66 settings.tuningParameters.entityRecoveryStrategy)
67
68 settings.tuningParameters.entityRecoveryStrategy match {
69 case "all" =>
70 idsLeftToStart = Set.empty
71 startBatch(ids)
```

Code Correctness: Non-Static Inner Class Implements Serializable (58 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

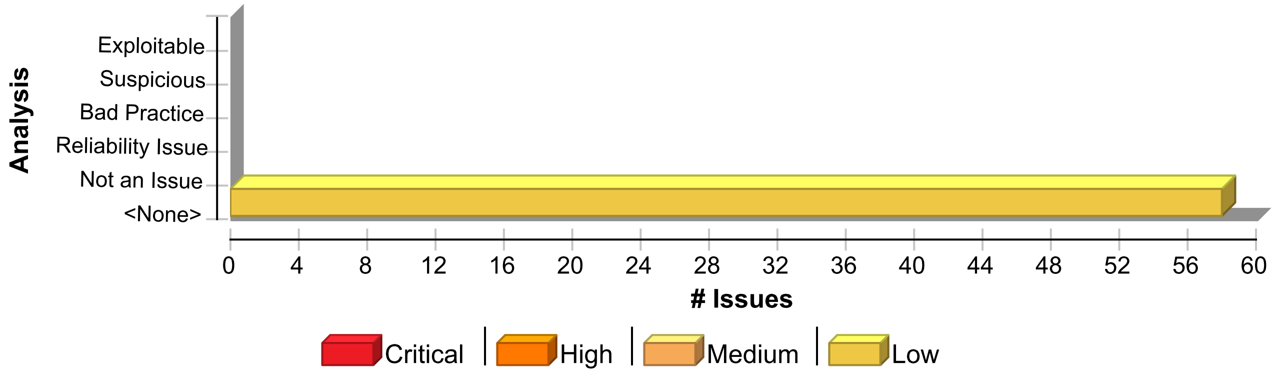
When using inner classes, they should not be serialized, or they should be changed to static-nested classes, since these do not have the drawbacks that non-static inner classes have when serialized. When a nested class is static it inherently has no association with instance variables (including those 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	58	0	0	58
Total	**58**	**0**	**0**	**58**

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

Package: akka.cluster.sharding

Shard.scala, line 113 (Code Correctness: Non-Static Inner Class Implements Serializable) Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$RememberedEntityIds
File: Shard.scala:113
Taint Flags:

```
110
111  private final case class EntityTerminated(ref: ActorRef)
112
113  private final case class RememberedEntityIds(ids: Set[EntityId])
114  private final case class RememberEntityStoreCrashed(store: ActorRef)
115
116  private val RememberEntityTimeoutKey = "RememberEntityTimeout"
```

ShardCoordinator.scala, line 526 (Code Correctness: Non-Static Inner Class Implements Serializable) Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardCoordinator$DelayedShardRegionTerminated
File: ShardCoordinator.scala:526
Taint Flags:

Code Correctness: Non-Static Inner Class Implements Serializable

Low

Package: akka.cluster.sharding

ShardCoordinator.scala, line 526 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

```
523 */
524 private final case class ResendShardHost(shard: ShardId, region: ActorRef)
525
526 private final case class DelayedShardRegionTerminated(region: ActorRef)
527
528 /**
529 * Result of `allocateShard` is piped to self with this message.
```

ShardRegion.scala, line 200 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$Passivate
File: ShardRegion.scala:200
Taint Flags:

```
197 *
198 * [[akka.actor.PoisonPill]] is a perfectly fine `stopMessage`.
199 */
200 @SerialVersionUID(1L) final case class Passivate(stopMessage: Any) extends ShardRegionCommand
201
202 /*
203 * Send this message to the `ShardRegion` actor to handoff all shards that are hosted by
```

ShardRegion.scala, line 269 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$GetClusterShardingStats
File: ShardRegion.scala:269
Taint Flags:

```
266 * Intended for testing purpose to see when cluster sharding is "ready" or to monitor
267 * the state of the shard regions.
268 */
269 @SerialVersionUID(1L) case class GetClusterShardingStats(timeout: FiniteDuration)
270 extends ShardRegionQuery
```

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

Package: akka.cluster.sharding

ShardRegion.scala, line 269 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
271  with ClusterShardingSerializable
272
```

ShardRegion.scala, line 497 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$SetActiveEntityLimit
File: ShardRegion.scala:497
Taint Flags:

```
494  * When limit-based automatic passivation is enabled, set a new active entity limit for a shard region.
495  */
496  @ApiMayChange
497  final case class SetActiveEntityLimit(perRegionLimit: Int)
498
499  /**
500  * INTERNAL API
```

ShardRegion.scala, line 474 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$RestartShard
File: ShardRegion.scala:474
Taint Flags:

```
471  * When an remembering entities and the shard stops unexpected (e.g. persist failure), we
472  * restart it after a back off using this message.
473  */
474  private final case class RestartShard(shardId: ShardId)
475
476  /**
477  * When remembering entities and a shard is started, each entity id that needs to
```

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

Package: akka.cluster.sharding

ShardRegion.scala, line 248 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$CurrentRegions
File: ShardRegion.scala:248
Taint Flags:

```
245  /**
246  * Reply to `GetCurrentRegions`
247  */
248  @SerialVersionUID(1L) final case class CurrentRegions(regions: Set[Address]) extends ClusterShardingSerializable {
249
250  /**
251  * Java API
```

ShardRegion.scala, line 506 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$ShardsUpdated
File: ShardRegion.scala:506
Taint Flags:

```
503  * Used for passivation strategies that change limits based on the number of active shards.
504  */
505  @InternalApi
506  private[akka] final case class ShardsUpdated(activeShards: Int) extends DeadLetterSuppression
507
508  /**
509  * INTERNAL API. Sends stopMessage (e.g. `PoisonPill`) to the entities and when all of
```

ShardCoordinator.scala, line 542 (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.sharding

ShardCoordinator.scala, line 542 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: ShardCoordinator$RebalanceWorker$ShardRegionTerminated
File: ShardCoordinator.scala:542
Taint Flags:

```
539 private final case class RebalanceResult(shards: Set[ShardId])
540
541 private[akka] object RebalanceWorker {
542 final case class ShardRegionTerminated(region: ActorRef)
543 }
544
545 /**
```

ShardCoordinator.scala, line 284 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardCoordinator$LeastShardAllocationStrategy
File: ShardCoordinator.scala:284
Taint Flags:

```
281 */
282 @SerialVersionUID(1L)
283 @DoNotInherit
284 class LeastShardAllocationStrategy(rebalanceThreshold: Int, maxSimultaneousRebalance: Int)
285 extends AbstractLeastShardAllocationStrategy
286 with Serializable {
287
```

Shard.scala, line 213 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$RememberingStart
File: Shard.scala:213
Taint Flags:

```
210 * its existence is being recorded in the remember entities store, or while the stop is queued up
211 * to be stored in the next batch.
212 */
```

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

Package: akka.cluster.sharding

Shard.scala, line 213 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
213 final case class RememberingStart(ackTo: Set[ActorRef]) extends EntityState {
214 override def transition(newState: EntityState, entities: Entities): EntityState = newState match {
215 case active: Active => active
216 case r: RememberingStart =>
```

ShardCoordinator.scala, line 539 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardCoordinator$RebalanceResult
File: ShardCoordinator.scala:539
Taint Flags:

```
536 /**
537 * Result of `rebalance` is piped to self with this message.
538 */
539 private final case class RebalanceResult(shards: Set[ShardId])
540
541 private[akka] object RebalanceWorker {
542 final case class ShardRegionTerminated(region: ActorRef)
```

ShardRegion.scala, line 277 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$ClusterShardingStats
File: ShardRegion.scala:277
Taint Flags:

```
274 * Reply to [[GetClusterShardingStats]], contains statistics about all the sharding regions
275 * in the cluster.
276 */
277 @SerialVersionUID(1L) final case class ClusterShardingStats(regions: Map[Address, ShardRegionStats])
278 extends ClusterShardingSerializable {
279
280 /**
```

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

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 668 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterShardingSettings$MostRecentlyUsedPassivationStrategy
File: ClusterShardingSettings.scala:668
Taint Flags:

```
665 * INTERNAL API
666 */
667 @InternalApi
668 private[akka] case class MostRecentlyUsedPassivationStrategy(limit: Int, idle: Option[IdlePassivationStrategy])
669 extends PassivationStrategy
670
671 /**
```

ClusterShardingSettings.scala, line 715 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterShardingSettings$CompositePassivationStrategy$FrequencySketchAdmissionFilter
File: ClusterShardingSettings.scala:715
Taint Flags:

```
712
713 case object AlwaysAdmissionFilter extends AdmissionFilter
714
715 case class FrequencySketchAdmissionFilter(
716 widthMultiplier: Int,
717 resetMultiplier: Double,
718 depth: Int,
```

Shard.scala, line 83 (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.sharding

Shard.scala, line 83 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: Shard$LeaseAcquireResult
File: Shard.scala:83
Taint Flags:

```
80 @SerialVersionUID(1L) final case class ShardStats(shardId: ShardRegion.ShardId, entityCount: Int)
81 extends ClusterShardingSerializable
82
83 final case class LeaseAcquireResult(acquired: Boolean, reason: Option[Throwable]) extends DeadLetterSuppression
84 final case class LeaseLost(reason: Option[Throwable]) extends DeadLetterSuppression
85
86 case object LeaseRetry extends DeadLetterSuppression
```

ShardCoordinator.scala, line 524 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardCoordinator$ResendShardHost
File: ShardCoordinator.scala:524
Taint Flags:

```
521 /**
522 * Check if we've received a shard start request
523 */
524 private final case class ResendShardHost(shard: ShardId, region: ActorRef)
525
526 private final case class DelayedShardRegionTerminated(region: ActorRef)
527
```

ShardCoordinator.scala, line 519 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardCoordinator$RebalanceDone
File: ShardCoordinator.scala:519
Taint Flags:

```
516 /**
517 * End of rebalance process performed by [[RebalanceWorker]]
518 */
```

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

Package: akka.cluster.sharding

ShardCoordinator.scala, line 519 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
519 private final case class RebalanceDone(shard: ShardId, ok: Boolean)
520
521 /**
522 * Check if we've received a shard start request
```

Shard.scala, line 114 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$RememberEntityStoreCrashed
File: Shard.scala:114
Taint Flags:

```
111 private final case class EntityTerminated(ref: ActorRef)
112
113 private final case class RememberedEntityIds(ids: Set[EntityId])
114 private final case class RememberEntityStoreCrashed(store: ActorRef)
115
116 private val RememberEntityTimeoutKey = "RememberEntityTimeout"
117 final case class RememberEntityTimeout(operation: RememberEntitiesShardStore.Command)
```

Shard.scala, line 111 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$EntityTerminated
File: Shard.scala:111
Taint Flags:

```
108
109 case object PassivateIntervalTick extends NoSerializationVerificationNeeded
110
111 private final case class EntityTerminated(ref: ActorRef)
112
113 private final case class RememberedEntityIds(ids: Set[EntityId])
114 private final case class RememberEntityStoreCrashed(store: ActorRef)
```

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

Package: akka.cluster.sharding

ShardRegion.scala, line 422 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$ShardState
File: ShardRegion.scala:422
Taint Flags:

```
419 Option(state.shards)
420 }
421
422 @SerialVersionUID(1L) final case class ShardState(shardId: ShardId, entityIds: Set[EntityId]) {
423
424 /**
425 * Java API:
```

ClusterSharding.scala, line 695 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterShardingGuardian$Started
File: ClusterSharding.scala:695
Taint Flags:

```
692 extractEntityId: ShardRegion.ExtractEntityId,
693 extractShardId: ShardRegion.ExtractShardId)
694 extends NoSerializationVerificationNeeded
695 final case class Started(shardRegion: ActorRef) extends NoSerializationVerificationNeeded
696 }
697
698 /**
```

ClusterShardingSettings.scala, line 687 (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.sharding

ClusterShardingSettings.scala, line 687 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: ClusterShardingSettings$LeastFrequentlyUsedPassivationStrategy
File: ClusterShardingSettings.scala:687
Taint Flags:

```
684 * INTERNAL API
685 */
686 @InternalApi
687 private[akka] case class LeastFrequentlyUsedPassivationStrategy(
688 limit: Int,
689 dynamicAging: Boolean,
690 idle: Option[IdlePassivationStrategy])
```

ShardRegion.scala, line 375 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$CurrentShardRegionState
File: ShardRegion.scala:375
Taint Flags:

```
372 *
373 * If gathering the shard information times out the set of shards will be empty.
374 */
375 @SerialVersionUID(1L) final class CurrentShardRegionState(val shards: Set[ShardState], val failed: Set[ShardId])
376 extends ClusterShardingSerializable
377 with Product {
378
```

RemoveInternalClusterShardingData.scala, line 126 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: RemoveInternalClusterShardingData$RemoveOnePersistenceId$Removals
File: RemoveInternalClusterShardingData.scala:126
Taint Flags:

```
123 Props(new RemoveOnePersistenceId(journalPluginId, persistenceId: String, replyTo))
124
```

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

Package: akka.cluster.sharding

RemoveInternalClusterShardingData.scala, line 126 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
125 case class Result(removals: Try[Removals])
126 case class Removals(events: Boolean, snapshots: Boolean)
127 }
128
129 /**
```

Shard.scala, line 76 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$CurrentShardState
File: Shard.scala:76
Taint Flags:

```
73
74 @SerialVersionUID(1L) case object GetCurrentShardState extends ShardQuery
75
76 @SerialVersionUID(1L) final case class CurrentShardState(shardId: ShardRegion.ShardId, entityIds: Set[EntityId])
77
78 @SerialVersionUID(1L) case object GetShardStats extends ShardQuery with ClusterShardingSerializable
79
```

Shard.scala, line 61 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$RestartTerminatedEntity
File: Shard.scala:61
Taint Flags:

```
58 * When remembering entities and the entity stops without issuing a `Passivate`, we
59 * restart it after a back off using this message.
60 */
61 final case class RestartTerminatedEntity(entity: EntityId) extends RememberEntityCommand
62
63 /**
64 * If the shard id extractor is changed, remembered entities will start in a different shard
```

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

Package: akka.cluster.sharding

Shard.scala, line 84 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$LeaseLost
File: Shard.scala:84
Taint Flags:

```
81  extends ClusterShardingSerializable
82
83  final case class LeaseAcquireResult(acquired: Boolean, reason: Option[Throwable]) extends DeadLetterSuppression
84  final case class LeaseLost(reason: Option[Throwable]) extends DeadLetterSuppression
85
86  case object LeaseRetry extends DeadLetterSuppression
87  private val LeaseRetryTimer = "lease-retry"
```

ClusterShardingSettings.scala, line 630 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterShardingSettings$IdlePassivationStrategy
File: ClusterShardingSettings.scala:630
Taint Flags:

```
627  * INTERNAL API
628  */
629  @InternalApi
630  private[akka] case class IdlePassivationStrategy(timeout: FiniteDuration, interval: FiniteDuration)
631  extends PassivationStrategy
632
633  /**
```

ClusterShardingSettings.scala, line 658 (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.sharding

ClusterShardingSettings.scala, line 658 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: ClusterShardingSettings$LeastRecentlyUsedPassivationStrategy
File: ClusterShardingSettings.scala:658
Taint Flags:

```
655 * INTERNAL API
656 */
657 @InternalApi
658 private[akka] case class LeastRecentlyUsedPassivationStrategy(
659 limit: Int,
660 segmented: immutable.Seq[Double],
661 idle: Option[IdlePassivationStrategy])
```

ClusterShardingSettings.scala, line 740 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterShardingSettings$CompositePassivationStrategy$HillClimbingAdmissionOptimizer
File: ClusterShardingSettings.scala:740
Taint Flags:

```
737
738 case object NoAdmissionOptimizer extends AdmissionOptimizer
739
740 case class HillClimbingAdmissionOptimizer(
741 adjustMultiplier: Double,
742 initialStep: Double,
743 restartThreshold: Double,
```

Shard.scala, line 251 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$Passivating
File: Shard.scala:251
Taint Flags:

```
248 case _ => invalidTransition(newState, entities)
249 }
250 }
```

Code Correctness: Non-Static Inner Class Implements Serializable

Low

Package: akka.cluster.sharding

Shard.scala, line 251 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

```
251 final case class Passivating(ref: ActorRef) extends WithRef {
252 override def transition(newState: EntityState, entities: Entities): EntityState = newState match {
253 case RememberingStop => RememberingStop
254 case NoState if !entities.rememberingEntities => NoState
```

RemoveInternalClusterShardingData.scala, line 125 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: RemoveInternalClusterShardingData$RemoveOnePersistenceId$Result
File: RemoveInternalClusterShardingData.scala:125
Taint Flags:

```
122 def props(journalPluginId: String, persistenceId: String, replyTo: ActorRef): Props =
123 Props(new RemoveOnePersistenceId(journalPluginId, persistenceId: String, replyTo))
124
125 case class Result(removals: Try[Removals])
126 case class Removals(events: Boolean, snapshots: Boolean)
127 }
128
```

Shard.scala, line 80 (Code Correctness: Non-Static Inner Class Implements Serializable)

Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$ShardStats
File: Shard.scala:80
Taint Flags:

```
77
78 @SerialVersionUID(1L) case object GetShardStats extends ShardQuery with ClusterShardingSerializable
79
80 @SerialVersionUID(1L) final case class ShardStats(shardId: ShardRegion.ShardId, entityCount: Int)
81 extends ClusterShardingSerializable
82
83 final case class LeaseAcquireResult(acquired: Boolean, reason: Option[Throwable]) extends DeadLetterSuppression
```

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

Package: akka.cluster.sharding

ShardRegion.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: ShardRegion$ShardInitialized
File: ShardRegion.scala:223
Taint Flags:

```
220  * We must be sure that a shard is initialized before to start send messages to it.
221  * Shard could be terminated during initialization.
222  */
223  final case class ShardInitialized(shardId: ShardId)
224
225  sealed trait ShardRegionQuery
226
```

Shard.scala, line 243 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$Active
File: Shard.scala:243
Taint Flags:

```
240  sealed trait WithRef extends EntityState {
241  def ref: ActorRef
242  }
243  final case class Active(ref: ActorRef) extends WithRef {
244  override def transition(newState: EntityState, entities: Entities): EntityState = newState match {
245  case passivating: Passivating => passivating
246  case WaitingForRestart => WaitingForRestart
```

ShardRegion.scala, line 481 (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.sharding

ShardRegion.scala, line 481 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: ShardRegion$StartEntity
File: ShardRegion.scala:481
Taint Flags:

```
478 * be running will trigger this message being sent through sharding. For this to work
479 * the message *must* be handled by the shard id extractor.
480 */
481 final case class StartEntity(entityId: EntityId) extends ClusterShardingSerializable
482
483 /**
484 * Sent back when a `ShardRegion.StartEntity` message was received and triggered the entity
```

Shard.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: Shard$EntitiesMovedToOtherShard
File: Shard.scala:67
Taint Flags:

```
64 * If the shard id extractor is changed, remembered entities will start in a different shard
65 * and this message is sent to the shard to not leak `entityId -> RememberedButNotStarted` entries
66 */
67 final case class EntitiesMovedToOtherShard(ids: Set[ShardRegion.ShardId]) extends RememberEntityCommand
68
69 /**
70 * A query for information about the shard
```

ShardRegion.scala, line 311 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardRegion$ShardRegionStats
File: ShardRegion.scala:311
Taint Flags:

```
308 * @param stats the region stats mapping of `ShardId` to number of entities
309 * @param failed set of shards if any failed to respond within the timeout
310 */
```

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

Package: akka.cluster.sharding

ShardRegion.scala, line 311 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
311  @SerialVersionUID(1L) final class ShardRegionStats(val stats: Map[ShardId, Int], val failed: Set[ShardId])
312  extends ClusterShardingSerializable
313  with Product {
314
```

ClusterSharding.scala, line 679 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterShardingGuardian$Start
File: ClusterSharding.scala:679
Taint Flags:

```
676  */
677 private[akka] object ClusterShardingGuardian {
678  import ShardCoordinator.ShardAllocationStrategy
679  final case class Start(
680  typeName: String,
681  entityProps: String => Props,
682  settings: ClusterShardingSettings,
```

Shard.scala, line 117 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: Shard$RememberEntityTimeout
File: Shard.scala:117
Taint Flags:

```
114  private final case class RememberEntityStoreCrashed(store: ActorRef)
115
116  private val RememberEntityTimeoutKey = "RememberEntityTimeout"
117  final case class RememberEntityTimeout(operation: RememberEntitiesShardStore.Command)
118
119  /**
120  * State machine for an entity:
```

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

Package: akka.cluster.sharding

ClusterSharding.scala, line 688 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterShardingGuardian$StartProxy
File: ClusterSharding.scala:688
Taint Flags:

```
685 allocationStrategy: ShardAllocationStrategy,
686 handOffStopMessage: Any)
687 extends NoSerializationVerificationNeeded
688 final case class StartProxy(
689 typeName: String,
690 dataCenter: Option[DataCenter],
691 settings: ClusterShardingSettings,
```

ShardCoordinator.scala, line 1429 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DDataShardCoordinator$RememberEntitiesTimeout
File: ShardCoordinator.scala:1429
Taint Flags:

```
1426
1427 private case object RememberEntitiesStoreStopped
1428
1429 private case class RememberEntitiesTimeout(shardId: ShardId)
1430
1431 private case object RememberEntitiesLoadTimeout
1432
```

ShardRegion.scala, line 487 (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.sharding

ShardRegion.scala, line 487 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: ShardRegion$StartEntityAck
File: ShardRegion.scala:487
Taint Flags:

```
484 * Sent back when a `ShardRegion.StartEntity` message was received and triggered the entity
485 * to start (it does not guarantee the entity successfully started)
486 */
487 final case class StartEntityAck(entityId: EntityId, shardId: ShardRegion.ShardId)
488 extends ClusterShardingSerializable
489 with DeadLetterSuppression
490
```

ShardCoordinator.scala, line 531 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardCoordinator$AllocateShardResult
File: ShardCoordinator.scala:531
Taint Flags:

```
528 /**
529 * Result of `allocateShard` is piped to self with this message.
530 */
531 private final case class AllocateShardResult(
532 shard: ShardId,
533 shardRegion: Option[ActorRef],
534 getShardHomeSender: ActorRef)
```

ClusterShardingSettings.scala, line 776 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterShardingSettings$CompositePassivationStrategy
File: ClusterShardingSettings.scala:776
Taint Flags:

```
773 * INTERNAL API
774 */
```

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

Package: akka.cluster.sharding

ClusterShardingSettings.scala, line 776 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
775 @InternalApi
776 private[akka] case class CompositePassivationStrategy(
777 limit: Int,
778 mainStrategy: PassivationStrategy,
779 windowStrategy: PassivationStrategy,
```

Package: akka.cluster.sharding.external

external/ExternalShardAllocationStrategy.scala, line 41 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ExternalShardAllocationStrategy$GetShardLocationResponse
File: external/ExternalShardAllocationStrategy.scala:41
Taint Flags:

```
38 private[akka] final case class GetShardLocation(shard: ShardId)
39 private[akka] case object GetShardLocations
40 private[akka] final case class GetShardLocationsResponse(desiredAllocations: Map[ShardId, Address])
41 private[akka] final case class GetShardLocationResponse(address: Option[Address])
42
43 // only returned locally, serialized as a string
44 final case class ShardLocation(address: Address) extends NoSerializationVerificationNeeded
```

external/ExternalShardAllocationStrategy.scala, line 40 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ExternalShardAllocationStrategy$GetShardLocationsResponse
File: external/ExternalShardAllocationStrategy.scala:40
Taint Flags:

```
37 // local only messages
38 private[akka] final case class GetShardLocation(shard: ShardId)
39 private[akka] case object GetShardLocations
40 private[akka] final case class GetShardLocationsResponse(desiredAllocations: Map[ShardId, Address])
41 private[akka] final case class GetShardLocationResponse(address: Option[Address])
42
```

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

Package: akka.cluster.sharding.external

external/ExternalShardAllocationStrategy.scala, line 40 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
43 // only returned locally, serialized as a string
```

external/ExternalShardAllocationStrategy.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: ExternalShardAllocationStrategy$GetShardLocation
File: external/ExternalShardAllocationStrategy.scala:38
Taint Flags:

```
35 type ShardRegion = ActorRef
36
37 // local only messages
38 private[akka] final case class GetShardLocation(shard: ShardId)
39 private[akka] case object GetShardLocations
40 private[akka] final case class GetShardLocationsResponse(desiredAllocations: Map[ShardId, Address])
41 private[akka] final case class GetShardLocationResponse(address: Option[Address])
```

external/ExternalShardAllocationStrategy.scala, line 44 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ExternalShardAllocationStrategy$ShardLocation
File: external/ExternalShardAllocationStrategy.scala:44
Taint Flags:

```
41 private[akka] final case class GetShardLocationResponse(address: Option[Address])
42
43 // only returned locally, serialized as a string
44 final case class ShardLocation(address: Address) extends NoSerializationVerificationNeeded
45
46 private object DDataStateActor {
47 def props(typeName: String) = Props(new DDataStateActor(typeName))
```

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

Package: akka.cluster.sharding.internal

internal/RememberEntityStarter.scala, line 36 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: RememberEntityStarter$StartBatch
File: internal/RememberEntityStarter.scala:36
Taint Flags:

```
33 settings: ClusterShardingSettings) =
34 Props(new RememberEntityStarter(region, shard, shardId, ids, settings))
35
36 private final case class StartBatch(batchSize: Int) extends NoSerializationVerificationNeeded
37 private case object ResendUnAcked extends NoSerializationVerificationNeeded
38 }
39
```

internal/EventSourcedRememberEntitiesShardStore.scala, line 44 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: EventSourcedRememberEntitiesShardStore$EntitiesStarted
File: internal/EventSourcedRememberEntitiesShardStore.scala:44
Taint Flags:

```
41 /**
42 * `State` change for starting a set of entities in this `Shard`
43 */
44 final case class EntitiesStarted(entities: Set[EntityId]) extends StateChange
45
46 case object StartedAck
47
```

internal/EventSourcedRememberEntitiesShardStore.scala, line 39 (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.sharding.internal

internal/EventSourcedRememberEntitiesShardStore.scala, line 39 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: EventSourcedRememberEntitiesShardStore$State
File: internal/EventSourcedRememberEntitiesShardStore.scala:39
Taint Flags:

```
36 /**
37 * Persistent state of the Shard.
38 */
39 final case class State private[akka] (entities: Set[EntityId] = Set.empty) extends ClusterShardingSerializable
40
41 /**
42 * `State` change for starting a set of entities in this `Shard`
```

internal/DDataRememberEntitiesShardStore.scala, line 66 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DDataRememberEntitiesShardStore$Stopped
File: internal/DDataRememberEntitiesShardStore.scala:66
Taint Flags:

```
63 def id: EntityId
64 }
65 private case class Started(id: EntityId) extends Evt
66 private case class Stopped(id: EntityId) extends Evt
67
68 }
69
```

internal/EventSourcedRememberEntitiesShardStore.scala, line 51 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: EventSourcedRememberEntitiesShardStore$EntitiesStopped
File: internal/EventSourcedRememberEntitiesShardStore.scala:51
Taint Flags:

```
48 /**
49 * `State` change for an entity which has terminated.
```

Code Correctness: Non-Static Inner Class Implements Serializable Low

Package: akka.cluster.sharding.internal

internal/EventSourcedRememberEntitiesShardStore.scala, line 51 (Code Correctness: Non-Static Inner Class Implements Serializable) Low

```
50 */
51 final case class EntitiesStopped(entities: Set[EntityId]) extends StateChange
52
53 def props(typeName: String, shardId: ShardRegion.ShardId, settings: ClusterShardingSettings): Props =
54 Props(new EventSourcedRememberEntitiesShardStore(typeName, shardId, settings))
```

internal/EventSourcedRememberEntitiesCoordinatorStore.scala, line 26 (Code Correctness: Non-Static Inner Class Implements Serializable) Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: EventSourcedRememberEntitiesCoordinatorStore$State
File: internal/EventSourcedRememberEntitiesCoordinatorStore.scala:26
Taint Flags:

```
23 def props(typeName: String, settings: ClusterShardingSettings): Props =
24 Props(new EventSourcedRememberEntitiesCoordinatorStore(typeName, settings))
25
26 case class State(shards: Set[ShardId], writtenMigrationMarker: Boolean = false) extends ClusterShardingSerializable
27
28 case object MigrationMarker extends ClusterShardingSerializable
29 }
```

internal/AbstractLeastShardAllocationStrategy.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: AbstractLeastShardAllocationStrategy$RegionEntry
File: internal/AbstractLeastShardAllocationStrategy.scala:38
Taint Flags:

```
35
36 type AllocationMap = Map[ActorRef, immutable.IndexedSeq[ShardId]]
37
38 final case class RegionEntry(region: ActorRef, member: Member, shardIds: immutable.IndexedSeq[ShardId])
39
40 implicit object ShardSuitabilityOrdering extends Ordering[RegionEntry] {
41 override def compare(x: RegionEntry, y: RegionEntry): Int = {
```

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

Package: akka.cluster.sharding.internal

internal/AbstractLeastShardAllocationStrategy.scala, line 38 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

internal/DDataRememberEntitiesShardStore.scala, line 65 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: DDataRememberEntitiesShardStore$Started
File: internal/DDataRememberEntitiesShardStore.scala:65
Taint Flags:

```
62  private sealed trait Evt {
63  def id: EntityId
64  }
65  private case class Started(id: EntityId) extends Evt
66  private case class Stopped(id: EntityId) extends Evt
67
68 }
```

J2EE Bad Practices: Leftover Debug Code (1 issue)

Abstract

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

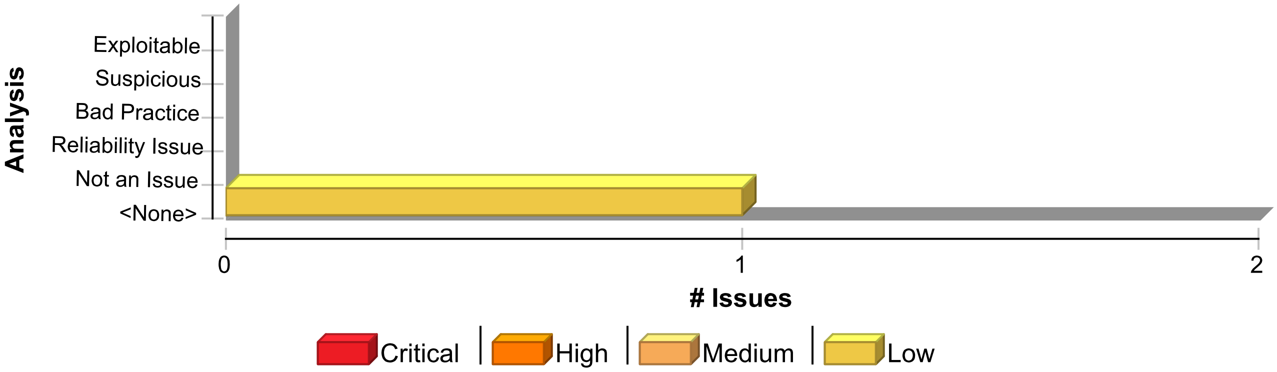
Explanation

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

Recommendation

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

Issue Summary



Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
J2EE Bad Practices: Leftover Debug Code	1	0	0	1
Total	**1**	**0**	**0**	**1**

J2EE Bad Practices: Leftover Debug Code — Low

Package: akka.cluster.sharding

RemoveInternalClusterShardingData.scala, line 62 (J2EE Bad Practices: Leftover Debug Code) — Low

Issue Details

Kingdom: Encapsulation
Scan Engine: SCA (Structural)

Sink Details

J2EE Bad Practices: Leftover Debug Code — Low

Package: akka.cluster.sharding

RemoveInternalClusterShardingData.scala, line 62 (J2EE Bad Practices: Leftover Debug Code) — Low

Sink: Function: main
Enclosing Method: main()
File: RemoveInternalClusterShardingData.scala:62
Taint Flags:

```
59 /**
60 * @see [[RemoveInternalClusterShardingData$ RemoveInternalClusterShardingData companion object]]
61 */
62 def main(args: Array[String]): Unit = {
63 if (args.isEmpty)
64 println("Specify the Cluster Sharding type names to remove in program arguments")
65 else {
```