Fortify Standalone Report Generator

Developer Workbook

akka-cluster-sharding-typed

Table of Contents

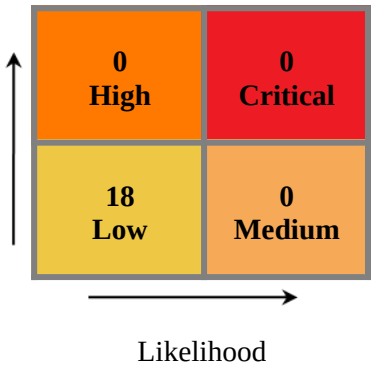
Executive Summary

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-typed 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-type(
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
WebInspect Agent:	Results Not Present
Other:	Results Not Present

Issues by Priority



Top Ten Critical Categories

This project does not contain any critical issues

Project Description

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

SCA

Date of Last Analysis:	Jun 16, 2022, 11:22 AM	**Engine Version:**	21.1.1.0009
Host Name:	Jacks-Work-MBP.local	**Certification:**	VALID
Number of Files:	26	**Lines of Code:**	1,666

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 / 3	0 / 3
Code Correctness: Non-Static Inner Class Implements Serializable	0	0	0	0 / 15	0 / 15

Results Outline

Code Correctness: Constructor Invokes Overridable Function (3 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");
    }
  }
}
```

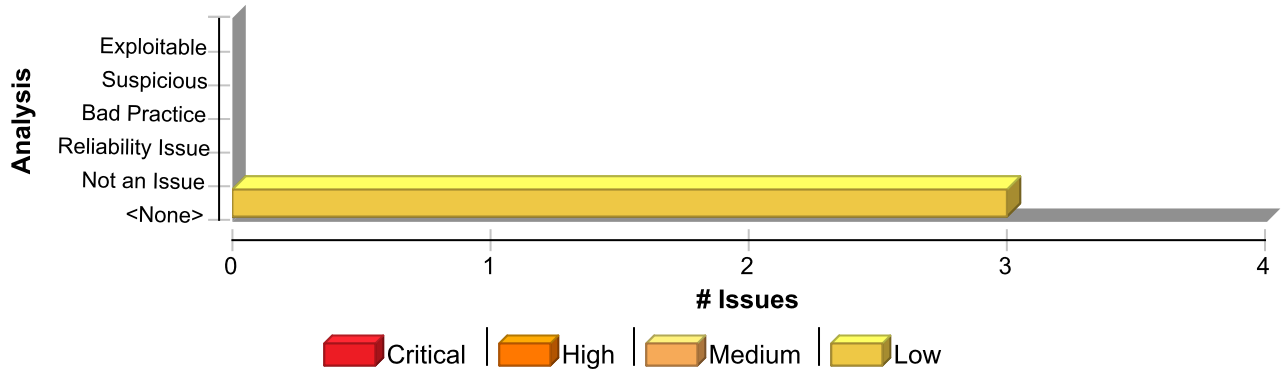
The code in `Example 1` prints "Attack successful!", since the `Attacker` class overrides the `validateUser()` function that is called from the constructor of the superclass `User`, and Java will first look in the subclass for functions called from the constructor.

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	3	0	0	3
Total	**3**	**0**	**0**	**3**

Code Correctness: Constructor Invokes Overridable Function	**Low**
Package: akka.cluster.sharding.typed.delivery.internal	
delivery/internal/ShardingProducerControllerImpl.scala, line 283 (Code Correctness: Constructor Invokes Overridable Function)	**Low**
Issue Details	

Code Correctness: Constructor Invokes Overridable Function — Low

Package: akka.cluster.sharding.typed.delivery.internal

delivery/internal/ShardingProducerControllerImpl.scala, line 283 (Code Correctness: Constructor Invokes Overridable Function) — Low

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: producerControllerSettings
Enclosing Method: ShardingProducerControllerImpl()
File: delivery/internal/ShardingProducerControllerImpl.scala:283
Taint Flags:

```
280 import ShardingProducerControllerImpl._
281
282 private val producerControllerSettings = settings.producerControllerSettings
283 private val durableQueueAskTimeout: Timeout = producerControllerSettings.durableQueueRequestTimeout
284 private val entityAskTimeout: Timeout = settings.internalAskTimeout
285 private val traceEnabled = context.log.isTraceEnabled
286
```

Package: akka.cluster.sharding.typed.internal

internal/ClusterShardingImpl.scala, line 105 (Code Correctness: Constructor Invokes Overridable Function) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: FunctionCall: classicSystem
Enclosing Method: ClusterShardingImpl()
File: internal/ClusterShardingImpl.scala:105
Taint Flags:

```
102
103 private val cluster = Cluster(system)
104 private val classicSystem: ExtendedActorSystem = system.toClassic.asInstanceOf[ExtendedActorSystem]
105 private val classicSharding = akka.cluster.sharding.ClusterSharding(classicSystem)
106 private val log: LoggingAdapter = Logging(classicSystem, classOf[scaladsl.ClusterSharding])
107
108 // typeKey.name to messageClassName
```

internal/ClusterShardingImpl.scala, line 106 (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.typed.internal	
internal/ClusterShardingImpl.scala, line 106 (Code Correctness: Constructor Invokes Overridable Function)	**Low**

Sink Details

Sink: FunctionCall: classicSystem
Enclosing Method: ClusterShardingImpl()
File: internal/ClusterShardingImpl.scala:106
Taint Flags:

```
103 private val cluster = Cluster(system)
104 private val classicSystem: ExtendedActorSystem = system.toClassic.asInstanceOf[ExtendedActorSystem]
105 private val classicSharding = akka.cluster.sharding.ClusterSharding(classicSystem)
106 private val log: LoggingAdapter = Logging(classicSystem, classOf[scaladsl.ClusterSharding])
107
108 // typeKey.name to messageClassName
109 private val regions: ConcurrentHashMap[String, String] = new ConcurrentHashMap
```

Code Correctness: Non-Static Inner Class Implements Serializable (15 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

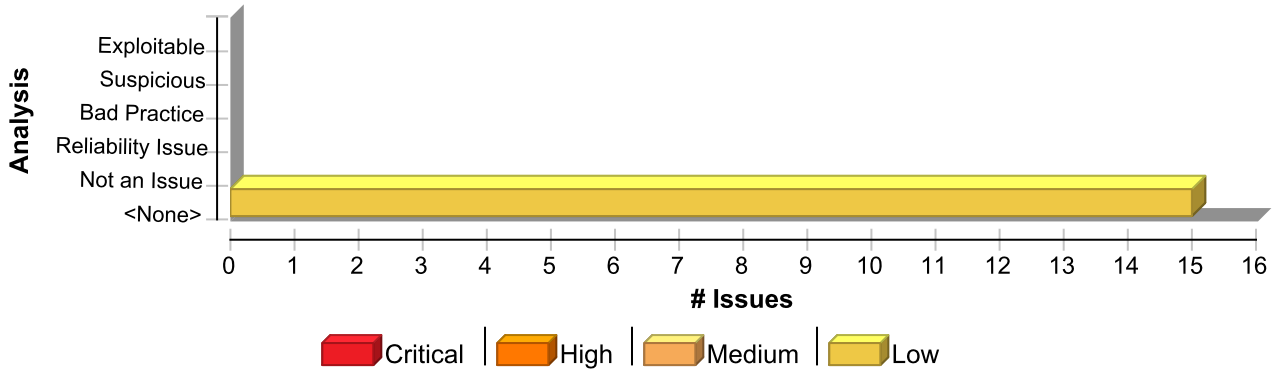
When using inner classes, they should not be serialized, or they should be changed to static-nested classes, since these do not have the drawbacks that non-static inner classes have when serialized. When a nested class is static it inherently has no association with instance variables (including those 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	15	0	0	15
Total	**15**	**0**	**0**	**15**

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

Package: akka.cluster.sharding.typed.delivery.internal

delivery/internal/ShardingProducerControllerImpl.scala, line 88 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardingProducerControllerImpl$State
File: delivery/internal/ShardingProducerControllerImpl.scala:88
Taint Flags:

```
85
86 private final case class Unconfirmed[A](totalSeqNr: TotalSeqNr, outSeqNr: OutSeqNr, replyTo: Option[ActorRef[Done]])
87
88 private final case class State[A](
89 currentSeqNr: TotalSeqNr,
90 producer: ActorRef[ShardingProducerController.RequestNext[A]],
91 out: Map[OutKey, OutState[A]],
```

delivery/internal/ShardingProducerControllerImpl.scala, line 52 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardingProducerControllerImpl$AskTimeout

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

Package: akka.cluster.sharding.typed.delivery.internal

delivery/internal/ShardingProducerControllerImpl.scala, line 52 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

File: delivery/internal/ShardingProducerControllerImpl.scala:52
Taint Flags:

```
49 private type OutKey = String
50
51 private final case class Ack(outKey: OutKey, confirmedSeqNr: OutSeqNr) extends InternalCommand
52 private final case class AskTimeout(outKey: OutKey, outSeqNr: OutSeqNr) extends InternalCommand
53
54 private final case class WrappedRequestNext[A](next: ProducerController.RequestNext[A]) extends InternalCommand
55
```

delivery/internal/ShardingProducerControllerImpl.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: ShardingProducerControllerImpl$Ack
File: delivery/internal/ShardingProducerControllerImpl.scala:51
Taint Flags:

```
48 private type OutSeqNr = Long
49 private type OutKey = String
50
51 private final case class Ack(outKey: OutKey, confirmedSeqNr: OutSeqNr) extends InternalCommand
52 private final case class AskTimeout(outKey: OutKey, outSeqNr: OutSeqNr) extends InternalCommand
53
54 private final case class WrappedRequestNext[A](next: ProducerController.RequestNext[A]) extends InternalCommand
```

delivery/internal/ShardingProducerControllerImpl.scala, line 63 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardingProducerControllerImpl$StoreMessageSentFailed
File: delivery/internal/ShardingProducerControllerImpl.scala:63
Taint Flags:

```
60 private case class LoadStateReply[A](state: DurableProducerQueue.State[A]) extends InternalCommand
61 private case class LoadStateFailed(attempt: Int) extends InternalCommand
62 private case class StoreMessageSentReply(ack: DurableProducerQueue.StoreMessageSentAck)
```

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

Package: akka.cluster.sharding.typed.delivery.internal

delivery/internal/ShardingProducerControllerImpl.scala, line 63 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
63 private case class StoreMessageSentFailed[A](messageSent: DurableProducerQueue.MessageSent[A], attempt: Int)
64 extends InternalCommand
65 private case class StoreMessageSentCompleted[A](messageSent: DurableProducerQueue.MessageSent[A])
66 extends InternalCommand
```

delivery/internal/ShardingProducerControllerImpl.scala, line 60 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardingProducerControllerImpl$LoadStateReply
File: delivery/internal/ShardingProducerControllerImpl.scala:60
Taint Flags:

```
57 def isAlreadyStored: Boolean = alreadyStored > 0
58 }
59
60 private case class LoadStateReply[A](state: DurableProducerQueue.State[A]) extends InternalCommand
61 private case class LoadStateFailed(attempt: Int) extends InternalCommand
62 private case class StoreMessageSentReply(ack: DurableProducerQueue.StoreMessageSentAck)
63 private case class StoreMessageSentFailed[A](messageSent: DurableProducerQueue.MessageSent[A], attempt: Int)
```

delivery/internal/ShardingProducerControllerImpl.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: ShardingProducerControllerImpl$StoreMessageSentCompleted
File: delivery/internal/ShardingProducerControllerImpl.scala:65
Taint Flags:

```
62 private case class StoreMessageSentReply(ack: DurableProducerQueue.StoreMessageSentAck)
63 private case class StoreMessageSentFailed[A](messageSent: DurableProducerQueue.MessageSent[A], attempt: Int)
64 extends InternalCommand
65 private case class StoreMessageSentCompleted[A](messageSent: DurableProducerQueue.MessageSent[A])
66 extends InternalCommand
67 private case object DurableQueueTerminated extends InternalCommand
68
```

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

Package: akka.cluster.sharding.typed.delivery.internal

delivery/internal/ShardingProducerControllerImpl.scala, line 86 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardingProducerControllerImpl$Unconfirmed
File: delivery/internal/ShardingProducerControllerImpl.scala:86
Taint Flags:

```
83
84 private final case class Buffered[A](totalSeqNr: TotalSeqNr, msg: A, replyTo: Option[ActorRef[Done]])
85
86 private final case class Unconfirmed[A](totalSeqNr: TotalSeqNr, outSeqNr: OutSeqNr, replyTo: Option[ActorRef[Done]])
87
88 private final case class State[A](
89 currentSeqNr: TotalSeqNr,
```

delivery/internal/ShardingProducerControllerImpl.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: ShardingProducerControllerImpl$Buffered
File: delivery/internal/ShardingProducerControllerImpl.scala:84
Taint Flags:

```
81 throw new IllegalStateException("nextTo and buffered shouldn't both be nonEmpty.")
82 }
83
84 private final case class Buffered[A](totalSeqNr: TotalSeqNr, msg: A, replyTo: Option[ActorRef[Done]])
85
86 private final case class Unconfirmed[A](totalSeqNr: TotalSeqNr, outSeqNr: OutSeqNr, replyTo: Option[ActorRef[Done]])
87
```

delivery/internal/ShardingProducerControllerImpl.scala, line 62 (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.typed.delivery.internal

delivery/internal/ShardingProducerControllerImpl.scala, line 62 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Sink: Class: ShardingProducerControllerImpl$StoreMessageSentReply
File: delivery/internal/ShardingProducerControllerImpl.scala:62
Taint Flags:

```
59
60 private case class LoadStateReply[A](state: DurableProducerQueue.State[A]) extends InternalCommand
61 private case class LoadStateFailed(attempt: Int) extends InternalCommand
62 private case class StoreMessageSentReply(ack: DurableProducerQueue.StoreMessageSentAck)
63 private case class StoreMessageSentFailed[A](messageSent: DurableProducerQueue.MessageSent[A], attempt: Int)
64 extends InternalCommand
65 private case class StoreMessageSentCompleted[A](messageSent: DurableProducerQueue.MessageSent[A])
```

delivery/internal/ShardingProducerControllerImpl.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: ShardingProducerControllerImpl$LoadStateFailed
File: delivery/internal/ShardingProducerControllerImpl.scala:61
Taint Flags:

```
58 }
59
60 private case class LoadStateReply[A](state: DurableProducerQueue.State[A]) extends InternalCommand
61 private case class LoadStateFailed(attempt: Int) extends InternalCommand
62 private case class StoreMessageSentReply(ack: DurableProducerQueue.StoreMessageSentAck)
63 private case class StoreMessageSentFailed[A](messageSent: DurableProducerQueue.MessageSent[A], attempt: Int)
64 extends InternalCommand
```

delivery/internal/ShardingProducerControllerImpl.scala, line 56 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardingProducerControllerImpl$Msg
File: delivery/internal/ShardingProducerControllerImpl.scala:56
Taint Flags:

```
53
54 private final case class WrappedRequestNext[A](next: ProducerController.RequestNext[A]) extends InternalCommand
```

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

Package: akka.cluster.sharding.typed.delivery.internal

delivery/internal/ShardingProducerControllerImpl.scala, line 56 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

```
55
56 private final case class Msg[A](envelope: ShardingEnvelope[A], alreadyStored: TotalSeqNr) extends InternalCommand {
57 def isAlreadyStored: Boolean = alreadyStored > 0
58 }
59
```

delivery/internal/ShardingProducerControllerImpl.scala, line 72 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardingProducerControllerImpl$OutState
File: delivery/internal/ShardingProducerControllerImpl.scala:72
Taint Flags:

```
69 private case object ResendFirstUnconfirmed extends InternalCommand
70 private case object CleanupUnused extends InternalCommand
71
72 private final case class OutState[A](
73 entityId: EntityId,
74 producerController: ActorRef[ProducerController.Command[A]],
75 nextTo: Option[ProducerController.RequestNext[A]],
```

delivery/internal/ShardingProducerControllerImpl.scala, line 54 (Code Correctness: Non-Static Inner Class Implements Serializable) — Low

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ShardingProducerControllerImpl$WrappedRequestNext
File: delivery/internal/ShardingProducerControllerImpl.scala:54
Taint Flags:

```
51 private final case class Ack(outKey: OutKey, confirmedSeqNr: OutSeqNr) extends InternalCommand
52 private final case class AskTimeout(outKey: OutKey, outSeqNr: OutSeqNr) extends InternalCommand
53
54 private final case class WrappedRequestNext[A](next: ProducerController.RequestNext[A]) extends InternalCommand
55
56 private final case class Msg[A](envelope: ShardingEnvelope[A], alreadyStored: TotalSeqNr) extends InternalCommand {
57 def isAlreadyStored: Boolean = alreadyStored > 0
```

Code Correctness: Non-Static Inner Class Implements Serializable	Low
Package: akka.cluster.sharding.typed.delivery.internal	
delivery/internal/ShardingProducerControllerImpl.scala, line 54 (Code Correctness: Non-Static Inner Class Implements Serializable)	**Low**

Package: akka.cluster.sharding.typed.javadsl	
javadsl/ClusterSharding.scala, line 52 (Code Correctness: Non-Static Inner Class Implements Serializable)	**Low**

Issue Details

Kingdom: Code Quality
Scan Engine: SCA (Structural)

Sink Details

Sink: Class: ClusterSharding$Passivate
File: javadsl/ClusterSharding.scala:52
Taint Flags:

```
49 * `stopMessage` message to the entity, which is then supposed to stop
50 * itself.
51 */
52 final case class Passivate[M](entity: ActorRef[M]) extends ShardCommand
53 }
54
55 /**
```

Package: akka.cluster.sharding.typed.scaladsl	
scaladsl/ClusterSharding.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: ClusterSharding$Passivate
File: scaladsl/ClusterSharding.scala:51
Taint Flags:

```
48 * `stopMessage` message to the entity, which is then supposed to stop
49 * itself.
50 */
51 final case class Passivate[M](entity: ActorRef[M]) extends ShardCommand with javadsl.ClusterSharding.ShardCommand
52
53 }
54
```