

# Fix malformed PUBLISH parsing and handle protocol errors safely #659



**HJianBo** merged 2 commits into `master` from `fix/publish-protocol-error-handli...`



on Mar 5



Conversation **1**



Commits **2**



Checks **5**



Files changed **8**



**HJianBo** commented on Mar 5 • edited

Member

Thanks tsuki <t5uki@.> to report this issue via email.

## Summary

- add strict bounds checks when parsing incoming `PUBLISH` topic bytes
- reject invalid MQTT 3.1.1 `PUBLISH` packets with zero-length topic instead of trapping
- align MQTT 5 behavior: allow empty topic only when Topic Alias is present
- on reader parse/protocol errors, disconnect the socket instead of continuing

## Why

Malformed packets could trigger a runtime trap during topic slicing ( `Range` requires `lowerBound <= upperBound` ), causing a remote crash. This change converts malformed input handling into safe parse failure plus connection close.

## Tests

- `swift test --filter FrameTests/testFramePublishRejectsZeroLengthTopic --filter FrameTests/testFramePublishRejectsZeroLengthTopicInMQTT5WithoutAlias --filter FrameTests/testFramePublishAllowsZeroLengthTopicInMQTT5WithAlias --filter CocoaMQTTReaderProtocolErrorTests`

Added tests:

- `testFramePublishRejectsZeroLengthTopic`
- `testFramePublishRejectsZeroLengthTopicInMQTT5WithoutAlias`
- `testFramePublishAllowsZeroLengthTopicInMQTT5WithAlias`
- `testMalformedPublishDisconnectsSocket`
- `testUnknownFrameTypeDisconnectsSocket`



[Fix malformed PUBLISH parsing and protocol error handling](#)

✖ [edd9a1d](#)



**HJianBo** requested a review from **Copilot** [last month](#)



**Copilot** [started reviewing](#) on behalf of **HJianBo** [last month](#)

[View session](#)



**Copilot** AI reviewed [on Mar 5](#)

[View reviewed changes](#)



**Copilot** AI left a comment

## Pull request overview

This PR fixes a remote crash vulnerability where malformed MQTT `PUBLISH` packets with zero-length topics could trigger a Swift runtime trap ( `Range requires lowerBound <= upperBound` ) during topic slicing. The fix adds proper bounds validation in `FramePublish` parsing and converts all parse/protocol errors in `CocoaMQTTReader` into safe disconnections rather than silent continuation.


### Changes:

- Added strict bounds checks and zero-length topic rejection in `FramePublish` initialization, with MQTT 5.0 allowing empty topics only when a Topic Alias is present
- Changed `CocoaMQTTReader.frameReady()` to disconnect the socket on any parse or protocol error instead of logging and continuing
- Added comprehensive test coverage for both the frame-level rejection and reader-level disconnect behavior

### Reviewed changes

Copilot reviewed 4 out of 4 changed files in this pull request and generated no comments.



File	Description
Source/FramePublish.swift	Added bounds checks, zero-length topic rejection for MQTT 3.1.1, and empty-topic-without-alias rejection for MQTT 5.0
Source/CocoaMQTTReader.swift	Replaced <code>printError + break</code> with <code>protocolError + return</code> across all frame parse failures; added <code>protocolError()</code> helper that logs and disconnects
CocoaMQTTTests/FrameTests.swift	Added three tests for zero-length topic handling across MQTT 3.1.1 and 5.0
CocoaMQTTTests/CocoaMQTTReaderProtocolErrorTests.swift	New test file with spy-based tests verifying socket disconnect on malformed publish and unknown frame type

 [Add Copilot custom instructions](#) for smarter, more guided reviews. [Learn how to get started.](#)





  [Handle MQTT5 DISCONNECT/AUTH frames and fix lint](#)

✓ [12ac4f9](#)

  **HJianBo** merged commit `010bca6` into `master` on [Mar 5](#)  
5 checks passed

[View details](#)

  **HJianBo** deleted the `fix/publish-protocol-error-handling` branch [last month](#)

[Sign up for free](#) to join this conversation on **GitHub**. Already have an account? [Sign in to comment](#)

Reviewers

 **Copilot**



Assignees

No one assigned

Labels

None yet

---

### Projects

None yet

---

### Milestone

No milestone

---

### Development

Successfully merging this pull request may close these issues.

None yet

---

### 2 participants

