

mtrudel / bandit Public

<> Code Issues 1 Pull requests 1 Actions Security and quality 5

Commit 1e8e559

mtrudel authored 6 hours ago · 23 / 27 · Verified

Merge commit from fork

main · 1.11.0

1 parent 45feea2 commit 1e8e559

2 files changed

+23 -15

↑ Top

Filter files...

lib/bandit/http2

frame.ex

test/bandit/http2

frame_deserialization_test.exs

Search within code

lib/bandit/http2/frame.ex

```

@@ -25,6 +25,20 @@ defmodule Bandit.HTTP2.Frame do
 25 25 | {:more, iodata()}, <<>>}
 26 26 | {:error, Bandit.HTTP2.Errors.error_code(), binary()}, iodata()}
 27 27 | nil
 28 + # This is a little more aggressive than necessary. RFC9113§4.2 says we only
    + # need
 29 + # to treat frame size violations as connection level errors if the frame in
 30 + # question would affect the connection as a whole, so we could be more
    + # surgical

```

```

31 + # here and send stream level errors in some cases. However, we are well within
32 + # our rights to consider such errors as connection errors
33 + def deserialize(
34 +     <<length::24, _type::8, _flags::8, _reserved::1, _stream_id::31,
    rest::binary>>,
35 +     max_frame_size
36 + )
37 + when length > max_frame_size do
38 +     {[:error, Bandit.HTTP2.Errors.frame_size_error(), "Payload size too large
    (RFC9113§4.2)"},
39 +     rest}
40 + end
41 +
28 42 def deserialize(
29 43     <<length::24, type::8, flags::8, _reserved::1, stream_id::31,
30 44     payload::binary-size(length), rest::binary>>,
    @@ -48,21 +62,6 @@ defmodule Bandit.HTTP2.Frame do
48 62     |> then(&{&1, rest})
49 63 end
50 64
51 - # This is a little more aggressive than necessary. RFC9113§4.2 says we only
    need
52 - # to treat frame size violations as connection level errors if the frame in
53 - # question would affect the connection as a whole, so we could be more
    surgical
54 - # here and send stream level errors in some cases. However, we are well within
55 - # our rights to consider such errors as connection errors
56 - def deserialize(
57 -     <<length::24, _type::8, _flags::8, _reserved::1, _stream_id::31,
58 -     _payload::binary-size(length), rest::binary>>,
59 -     max_frame_size
60 - )
61 - when length > max_frame_size do
62 -     {[:error, Bandit.HTTP2.Errors.frame_size_error(), "Payload size too large
    (RFC9113§4.2)"},
63 -     rest}
64 - end
65 -
66 65 # nil is used to indicate for Stream.unfold/2 that the frame deserialization
    is finished

```

```
67 66     def deserialize(<<>>, _max_frame_size) do
68 67         nil
```



...bandit/http2/frame_deserialization_test.exs



```
@@ -17,6 +17,15 @@ defmodule HTTP2FrameDeserializationTest do
```

```
17 17     end
18 18     end
19 19
20 +   describe "oversize frames" do
21 +     test "drops oversize frames early" do
22 +       frame = <<16_385::24, 0, 0, 0, 0, 0, 1>>
23 +
24 +       assert Frame.deserialize(frame, 16_384) ==
25 +         {[:error, 6, "Payload size too large (RFC9113§4.2)"}, <<>>}
26 +     end
27 +   end
28 +
20 29     describe "extra data" do
21 30       test "returns extra data" do
22 31         frame = <<0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 3>>
```



Comments 0



Please [sign in](#) to comment.