```
From ced205f0805027e9d9c0731f8c40b104220604ed Mon Sep 17 00:00:00 2001
From: Tatsuhiko Miyagawa <miyagawa@bulknews.net>
Date: Mon, 27 Apr 2026 12:25:05 -0700
Subject: [PATCH] Fix request smuggling (CVE-2026-40560): Transfer-Encoding
 takes precedence over Content-Length
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit

Per RFC 7230 §3.3.3, when both Transfer-Encoding and Content-Length are
present in a request, Transfer-Encoding must override Content-Length.
The previous if/elsif order allowed an attacker to supply a misleading
Content-Length alongside Transfer-Encoding: chunked, which could enable
HTTP request smuggling via a front-end reverse proxy.

Co-Authored-By: Claude Sonnet 4.6 <noreply@anthropic.com>
---
 Changes               |  2 ++
 lib/Starman/Server.pm | 28 ++++++++++-----------
 t/te_cl_precedence.t  | 57 +++++++++++++++++++++++++++++++++++++++++++
 3 files changed, 73 insertions(+), 14 deletions(-)
 create mode 100644 t/te_cl_precedence.t

diff --git a/Changes b/Changes
index e7692c8..462d613 100644
--- a/Changes
+++ b/Changes
@@ -1,6 +1,8 @@
 Revision history for Perl extension Starman

 {{$NEXT}}
+        - Fix HTTP request smuggling: Transfer-Encoding now takes precedence
+          over Content-Length per RFC 7230 §3.3.3 (CVE-2026-40560)

 0.4017  2023-09-13 13:27:02 PDT
         - Handle EINTR when doing sysread calls (Rob Mueller) #148
diff --git a/lib/Starman/Server.pm b/lib/Starman/Server.pm
index 14b4c53..7386e72 100644
--- a/lib/Starman/Server.pm
+++ b/lib/Starman/Server.pm
@@ -415,20 +415,7 @@ sub _prepare_env {

     my $chunked = do { no warnings; lc delete $env->{HTTP_TRANSFER_ENCODING} eq 'chunked' };

-    if (my $cl = $env->{CONTENT_LENGTH}) {
-        my $buf = Plack::TempBuffer->new($cl);
-        while ($cl > 0) {
-            my($chunk, $read) = $get_chunk->();
-
-            if ( !defined $read || $read == 0 ) {
-                die "Read error: $!\n";
-            }
-
-            $cl -= $read;
-            $buf->print($chunk);
-        }
-        $env->{'psgi.input'} = $buf->rewind;
-    } elsif ($chunked) {
+    if ($chunked) {
         my $buf = Plack::TempBuffer->new;
         my $chunk_buffer = '';
         my $length;
@@ -460,6 +447,19 @@ sub _prepare_env {
```

```
          $env->{CONTENT_LENGTH} = $length;
          $env->{'psgi.input'}   = $buf->rewind;
+     } elsif (my $cl = $env->{CONTENT_LENGTH}) {
+         my $buf = Plack::TempBuffer->new($cl);
+         while ($cl > 0) {
+             my($chunk, $read) = $get_chunk->();
+
+             if ( !defined $read || $read == 0 ) {
+                 die "Read error: $!\n";
+             }
+
+             $cl -= $read;
+             $buf->print($chunk);
+         }
+         $env->{'psgi.input'} = $buf->rewind;
      } else {
          $env->{'psgi.input'} = $null_io;
      }
diff --git a/t/te_cl_precedence.t b/t/te_cl_precedence.t
new file mode 100644
index 0000000..5e61c4c
--- /dev/null
+++ b/t/te_cl_precedence.t
@@ -0,0 +1,57 @@
+use strict;
+use warnings;
+use Test::TCP;
+use IO::Socket::INET qw/ SHUT_WR /;
+use HTTP::Response;
+use Plack::Loader;
+use Test::More;
+
+# RFC 7230 §3.3.3: when both Transfer-Encoding and Content-Length are
+# present, Transfer-Encoding must override Content-Length.
+test_tcp(
+    client => sub {
+        my $port = shift;
+
+        my $socket = IO::Socket::INET->new(
+            PeerAddr => 'localhost',
+            PeerPort => $port,
+            Proto    => 'tcp',
+        ) or die "Failed to connect: $!";
+
+        # Chunked body encodes "Hello World" (0xb = 11 bytes).
+        # Content-Length: 5 is intentionally wrong — it must be ignored.
+        my $chunked_body = "b\r\nHello World\r\n0\r\n\r\n";
+        my $req = "POST / HTTP/1.1\r\n"
+                . "Host: localhost\r\n"
+                . "Transfer-Encoding: chunked\r\n"
+                . "Content-Length: 5\r\n"
+                . "\r\n"
+                . $chunked_body;
+
+        $socket->send($req);
+        $socket->shutdown(SHUT_WR);
+
+        my $response = '';
+        while (1) {
+            my $n = $socket->sysread(my $buf, 4096);
+            last unless $n;
+            $response .= $buf;
+        }
+
```

```
+        my $res = HTTP::Response->parse($response);
+        is $res->content, 'Hello World',
+            'Transfer-Encoding: chunked takes precedence over Content-Length';
+    },
+    server => sub {
+        my $port = shift;
+        my $server = Plack::Loader->load('Starman', port => $port, host => '127.0.0.1');
+        $server->run(sub {
+            my $env = shift;
+            my $body = '';
+            $env->{'psgi.input'}->read($body, 8192);
+            return [ 200, [ 'Content-Type', 'text/plain', 'Content-Length', length($body)
], [ $body ] ];
+        });
+    },
+);
+
+done_testing;
```