⚠ This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

Commit c28c4a8

Browse files

ioquatix committed 4 days ago · 22 / 22 · Verified

```
Don't allow session to be recreated accidentally.
```

1 parent c837aff commit c28c4a8

Filter files...

- lib/rack/session
  - pool.rb
  - releases.md
- test
  - spec_session_pool.rb

3 files changed +59 -1 lines changed

Search within code

lib/rack/session/pool.rb +7 -1

```
@@ -53,6 +53,7 @@ def find_session(req, sid)
53  53
54  54          def write_session(req, session_id, new_session, options)
55  55            @mutex.synchronize do
    56  +           return false unless get_session_with_fallback(session_id)
56  57            @pool.store session_id.private_id, new_session
57  58            session_id
58  59            end
@@ -62,7 +63,12 @@ def delete_session(req, session_id, options)
62  63            @mutex.synchronize do
63  64            @pool.delete(session_id.public_id)
64  65            @pool.delete(session_id.private_id)
```

```
65      -           generate_sid(use_mutex: false) unless options[:drop]
    66  +
    67  +           unless options[:drop]
    68  +             sid = generate_sid(use_mutex: false)
    69  +             @pool.store(sid.private_id, {})
    70  +             sid
    71  +           end
66  72            end
67  73          end
68  74
```

releases.md +4

```
    ...     @@ -1,5 +1,9 @@
1   1       # Releases
2   2
    3   + ## Unreleased
    4   +
    5   +   - Prevent `Rack::Session::Pool` from recreating deleted sessions [CVE-2025-46336]
            (https://github.com/rack/rack-session/security/advisories/GHSA-9j94-67jr-4cqj).
    6   +
3   7       ## v2.1.0
4   8
5   9         - Improved compatibility with Ruby 3.3+ and Rack 3+.
```

test/spec_session_pool.rb +48

```
        @@ -288,4 +288,52 @@
288 288         res = Rack::MockRequest.new(app).get("/")
289 289         res["Set-Cookie"].must_be_nil
290 290       end
    291 +
    292 +   user_id_session = Rack::Lint.new(lambda do |env|
    293 +     session = env["rack.session"]
    294 +
    295 +     case env["PATH_INFO"]
    296 +     when "/login"
    297 +       session[:user_id] = 1
    298 +     when "/logout"
    299 +       if session[:user_id].nil?
    300 +         raise "User not logged in"
    301 +       end
    302 +
```

```
303 +       session.delete(:user_id)
304 +       session.options[:renew] = true
305 +     when "/slow"
306 +       Fiber.yield
307 +     end
308 +
309 +     Rack::Response.new(session.inspect).to_a
310 +   end)
311 +
312 +   it "doesn't allow session id to be reused" do
313 +     app = Rack::Session::Pool.new(user_id_session)
314 +
315 +     login_response = Rack::MockRequest.new(app).get("/login")
316 +     login_cookie = login_response["Set-Cookie"]
317 +
318 +     slow_request = Fiber.new do
319 +       Rack::MockRequest.new(app).get("/slow", "HTTP_COOKIE" => login_cookie)
320 +     end
321 +     slow_request.resume
322 +
323 +     # Check that the session is valid:
324 +     response = Rack::MockRequest.new(app).get("/", "HTTP_COOKIE" => login_cookie)
325 +     response.body.must_equal({"user_id" => 1}.to_s)
326 +
327 +     logout_response = Rack::MockRequest.new(app).get("/logout", "HTTP_COOKIE" =>
      login_cookie)
328 +     logout_cookie = logout_response["Set-Cookie"]
329 +
330 +     # Check that the session id is different after logout:
331 +     login_cookie[session_match].wont_equal logout_cookie[session_match]
332 +
333 +     slow_response = slow_request.resume
334 +
335 +     # Check that the cookie can't be reused:
336 +     response = Rack::MockRequest.new(app).get("/", "HTTP_COOKIE" => login_cookie)
337 +     response.body.must_equal "{}"
338 +   end
291 339   end
```

Comments 0

Please sign in to comment.