Arvados - Bug #17566

[controller] Request body size incorrectly limited to 10M when content-type is x-www-form-urlencoded

04/19/2021 10:00 PM - Tom Clegg

Status:	Resolved	**Start date:**	04/20/2021
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:	API	**Estimated time:**	0.00 hour
Target version:	2021-04-28 bughunt sprint		

Description

Subtasks:

Task # 17567: Review 17566-max-request-size	**Resolved**

Associated revisions

Revision 3fa0a550 - 04/21/2021 08:25 PM - Tom Clegg

Merge branch '17566-max-request-size'

fixes #17566

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tom@curii.com>

History

#2 - 04/20/2021 02:10 PM - Tom Clegg

- Release set to 39

We had a bug report that crunch-run logged "error in CaptureOutput: error creating output collection: unexpected EOF"; the controller log indicated a "create collection" call had a large request body and a 200 response code; and the RailsAPI log indicated an empty collection had been created successfully.

By default, (*http.Request)ParseForm() returns an error (without parsing the request body) if the request has content type "x-www-form-urlencoded" and the body exceeds 10 MB.

The "new" controller code path did not check for errors from ParseForm() when checking for "_method=GET" in the post vars. ParseForm() was called again later when obtaining the request parameters, and the error was checked -- but when ParseForm() has already returned an error, calling it again returns success, so the error was still unreported and the request was treated as if the body had been empty. This is why the request sent from controller to RailsAPI was "create empty collection", and controller logged a 200 response.

When ParseForm rejects an oversize body, it also notifies the http server code, so it closes the connection instead of mixing up the protocol by failing to read the entire request body. This is why the client (crunch-run) code saw an "unexpected EOF" error instead of a new (unexpectedly empty) collection record.

Fixes:

- Check the ParseForm error
- Use the server config (API.MaxRequestSize) instead of the default 10M request body size limit
- Return 413 when a POST form body exceeds the configured limit
- Add tests in lib/controller/router (where the request size limit is implemented) and sdk/go/arvadosclient
- Don't call StartAPI() from arvadosclient tests (this inadvertently caused the tests to bypass controller and communicate with RailsAPI directly)

17566-max-request-size @ 3cf194da907fb95f9534268af4905c6db8a71b01 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/2431/

#3 - 04/20/2021 02:44 PM - Nico César

review 17566-max-request-size 3cf194da907fb95f9534268af4905c6db8a71b01

In sdk/go/arvadosclient/arvadosclient_test.go

Any reason for this?

```
@@ -28,14 +30,12 @@ var _ = Suite(&MockArvadosServerSuite{})
```

```
 type ServerRequiredSuite struct{}

 func (s *ServerRequiredSuite) SetUpSuite(c *C) {
-       arvadostest.StartAPI()
        arvadostest.StartKeep(2, false)
        RetryDelay = 0
 }

 func (s *ServerRequiredSuite) TearDownSuite(c *C) {
        arvadostest.StopKeep(2)
-       arvadostest.StopAPI()
 }
```

why nginx.conf had to be changed in sdk/python/tests/nginx.conf to `client_max_body_size 0;`? This will affect other tests in the future I think we shall we include something in the documentation as recommended config

Since we have 128m in doc/install/install-api-server.html.textile.liquid:

```
  # This value effectively limits the size of API objects users can
  # create, especially collections.  If you change this, you should
  # also ensure the following settings match it:
  # * `client_max_body_size` in the previous server section
  # * `API.MaxRequestSize` in config.yml
  client_max_body_size 128m;
```

is unclear to me what should we recommend our users (match them both to 128m? nginx config to 0 now that we can handle it gracerfully?)

#4 - 04/20/2021 04:15 PM - Tom Clegg

Nico César wrote:

> Any reason for this?
> [...]

Yes, this is: "Don't call StartAPI() from arvadosclient tests (this inadvertently caused the tests to bypass controller and communicate with RailsAPI directly)"

The test case didn't catch the bug until I did that. This probably deserves a more thorough fix though, since AFAIK bypassing controller is not desirable anywhere else, either.

> why nginx.conf had to be changed in sdk/python/tests/nginx.conf to `client_max_body_size 0;`? This will affect other tests in the future I think we shall we include something in the documentation as recommended config

This lets us send a large request body through Nginx during tests. The previous config (default Nginx size limit = 1M) made it impossible to hit the default Go size limit of 10M from a test case.

> Since we have 128m in doc/install/install-api-server.html.textile.liquid:
> [...]
>
> is unclear to me what should we recommend our users (match them both to 128m? nginx config to 0 now that we can handle it gracerfully?)

I think the documentation at https://doc.arvados.org/main/install/install-api-server.html is still correct:

```
  # This value effectively limits the size of API objects users can
  # create, especially collections.  If you change this, you should
  # also ensure the following settings match it:
  # * `client_max_body_size` in the previous server section
  # * `API.MaxRequestSize` in config.yml
  client_max_body_size 128m;
```

But in our test suite, we don't want to test the Nginx size limit feature, we just want it out of the way so we can exercise our own limits.

#5 - 04/21/2021 07:53 PM - Nico César

Good points.

I think is Ready to merge.

#6 - 04/21/2021 08:36 PM - Tom Clegg

- Status changed from In Progress to Resolved

Applied in changeset arvados|3fa0a5500d5b58b5d0b9ea940dac85be2da079ec.

#7 - 04/22/2021 03:55 PM - Tom Clegg

- Release changed from 39 to 38