Commit 7691276

dprince committed on Jan 22, 2013

```
Limit the size of HTTP requests.

Adds a new RequestBodySizeLimiter middleware to guard against
really large HTTP requests. The default max request size is 112k
although this limit is configurable via the 'max_request_body_size'
config parameter.

Fixes LP Bug #1099025.

Change-Id: Id51be3d9a0d829d63d55a92dca61a39a17629785
```

master · zed-eom ••• 8.0.0a0

1 parent 8748cfa commit 7691276

6 files changed +127 -5 lines changed

Top

Filter files…

- etc
  - keystone.conf.sample
- keystone
  - common
    - utils.py
    - config.py
    - exception.py
  - middleware
    - core.py
- tests
  - test_sizelimit.py

6 files changed +127 -5 lines changed

Search within code

etc/keystone.conf.sample

```
@@ -186,6 +186,9 @@ paste.filter_factory = keystone.contrib.s3:S3Extension.factory
186 186  [filter:url_normalize]
187 187  paste.filter_factory = keystone.middleware:NormalizingFilter.factory
188 188
    189 + [filter:sizelimit]
    190 + paste.filter_factory = keystone.middleware:RequestBodySizeLimiter.factory
    191 +
189 192  [filter:stats_monitoring]
190 193  paste.filter_factory = keystone.contrib.stats:StatsMiddleware.factory
191 194
@@ -202,13 +205,13 @@ paste.app_factory = keystone.service:v3_app_factory
202 205  paste.app_factory = keystone.service:admin_app_factory
203 206
204 207  [pipeline:public_api]
205     - pipeline = stats_monitoring url_normalize token_auth admin_token_auth xml_body json_body debug ec2_extension user_crud_extension public_service
    208 + pipeline = sizelimit stats_monitoring url_normalize token_auth admin_token_auth xml_body json_body debug ec2_extension user_crud_extension public_service
206 209
207 210  [pipeline:admin_api]
208     - pipeline = stats_monitoring url_normalize token_auth admin_token_auth xml_body json_body debug stats_reporting ec2_extension s3_extension crud_extension admin_service
    211 + pipeline = sizelimit stats_monitoring url_normalize token_auth admin_token_auth xml_body json_body debug stats_reporting ec2_extension s3_extension crud_extension admin_service
209 212
210 213  [pipeline:api_v3]
211     - pipeline = stats_monitoring url_normalize token_auth admin_token_auth xml_body json_body debug stats_reporting ec2_extension s3_extension service_v3
    214 + pipeline = sizelimit stats_monitoring url_normalize token_auth admin_token_auth xml_body json_body debug stats_reporting ec2_extension s3_extension service_v3
212 215
213 216  [app:public_version_service]
214 217  paste.app_factory = keystone.service:public_version_app_factory
```

```
@@ -217,10 +220,10 @@ paste.app_factory =
keystone.service:public_version_app_factory
217 220 paste.app_factory = keystone.service:admin_version_app_factory
218 221
219 222 [pipeline:public_version_api]
220 - pipeline = stats_monitoring url_normalize xml_body public_version_service
223 + pipeline = sizelimit stats_monitoring url_normalize xml_body
public_version_service
221 224
222 225 [pipeline:admin_version_api]
223 - pipeline = stats_monitoring url_normalize xml_body admin_version_service
226 + pipeline = sizelimit stats_monitoring url_normalize xml_body
admin_version_service
224 227
225 228 [composite:main]
226 229 use = egg:Paste#urlmap
```

keystone/common/utils.py

```
@@ -311,3 +311,37 @@ def setup_remote_pydev_debug():
311 311 except:
312 312 LOG.exception(_(error_msg))
313 313 raise
314 +
315 +
316 + class LimitingReader(object):
317 + """Reader to limit the size of an incoming request."""
318 + def __init__(self, data, limit):
319 + """
320 + :param data: Underlying data object
321 + :param limit: maximum number of bytes the reader should allow
322 + """
323 + self.data = data
324 + self.limit = limit
325 + self.bytes_read = 0
326 +
327 + def __iter__(self):
328 + for chunk in self.data:
329 + self.bytes_read += len(chunk)
330 + if self.bytes_read > self.limit:
```

```
331 +                 raise exception.RequestTooLarge()
332 +             else:
333 +                 yield chunk
334 +
335 +     def read(self, i):
336 +         result = self.data.read(i)
337 +         self.bytes_read += len(result)
338 +         if self.bytes_read > self.limit:
339 +             raise exception.RequestTooLarge()
340 +         return result
341 +
342 +     def read(self):
343 +         result = self.data.read()
344 +         self.bytes_read += len(result)
345 +         if self.bytes_read > self.limit:
346 +             raise exception.RequestTooLarge()
347 +         return result
```

keystone/config.py

```
@@ -137,6 +137,8 @@ def register_cli_int(*args, **kw):
137 137  register_str('auth_admin_prefix', default='')
138 138  register_str('policy_file', default='policy.json')
139 139  register_str('policy_default_rule', default=None)
140 +#default max request size is 112k
141 +register_int('max_request_body_size', default=114688)
140 142
141 143  #ssl options
142 144  register_bool('enable', group='ssl', default=False)
```

keystone/exception.py

```
@@ -173,6 +173,12 @@ class Conflict(Error):
173 173      title = 'Conflict'
174 174
175 175
176 +class RequestTooLarge(Error):
177 +    """Request is too large."""
178 +    code = 413
179 +    title = 'Request is too large.'
```

```
      180 +
      181 +
176   182     class UnexpectedError(Error):
177   183         """An unexpected error prevented the server from fulfilling your request.
178   184
```

keystone/middleware/core.py

```
      @@ -14,7 +14,10 @@
14    14      # License for the specific language governing permissions and limitations
15    15      # under the License.
16    16
      17  +   import webob.dec
      18  +
17    19      from keystone.common import serializer
      20  +   from keystone.common import utils
18    21      from keystone.common import wsgi
19    22      from keystone import config
20    23      from keystone import exception
      @@ -164,3 +167,21 @@ def process_request(self, request):
164   167             # Rewrites path to root if no path is given.
165   168             elif not request.environ['PATH_INFO']:
166   169                 request.environ['PATH_INFO'] = '/'
      170 +
      171 +
      172 + class RequestBodySizeLimiter(wsgi.Middleware):
      173 +     """Limit the size of an incoming request."""
      174 +
      175 +     def __init__(self, *args, **kwargs):
      176 +         super(RequestBodySizeLimiter, self).__init__(*args, **kwargs)
      177 +
      178 +     @webob.dec.wsgify(RequestClass=wsgi.Request)
      179 +     def __call__(self, req):
      180 +
      181 +         if req.content_length > CONF.max_request_body_size:
      182 +             raise exception.RequestTooLarge()
      183 +         if req.content_length is None and req.is_body_readable:
      184 +             limiter = utils.LimitingReader(req.body_file,
      185 +                                            CONF.max_request_body_size)
```

```
186 +             req.body_file = limiter
187 +         return self.application
```

tests/test_sizelimit.py

```
... @@ -0,0 +1,56 @@
1  + # Copyright (c) 2013 OpenStack, LLC
2  + #
3  + #    Licensed under the Apache License, Version 2.0 (the "License"); you may
4  + #    not use this file except in compliance with the License. You may obtain
5  + #    a copy of the License at
6  + #
7  + #         http://www.apache.org/licenses/LICENSE-2.0
8  + #
9  + #    Unless required by applicable law or agreed to in writing, software
10 + #    distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
11 + #    WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
12 + #    License for the specific language governing permissions and limitations
13 + #    under the License.
14 +
15 + import webob
16 +
17 + from keystone import config
18 + from keystone import exception
19 + from keystone import middleware
20 + from keystone import test
21 +
22 + CONF = config.CONF
23 + MAX_REQUEST_BODY_SIZE = CONF.max_request_body_size
24 +
25 +
26 + class TestRequestBodySizeLimiter(test.TestCase):
27 +
28 +     def setUp(self):
29 +         super(TestRequestBodySizeLimiter, self).setUp()
30 +
31 +         @webob.dec.wsgify()
32 +         def fake_app(req):
33 +             return webob.Response(req.body)
34 +
35 +         self.middleware = middleware.RequestBodySizeLimiter(fake_app)
```

```
36 +         self.request = webob.Request.blank('/', method='POST')
37 +
38 +     def test_content_length_acceptable(self):
39 +         self.request.headers['Content-Length'] = MAX_REQUEST_BODY_SIZE
40 +         self.request.body = "0" * MAX_REQUEST_BODY_SIZE
41 +         response = self.request.get_response(self.middleware)
42 +         self.assertEqual(response.status_int, 200)
43 +
44 +     def test_content_length_too_large(self):
45 +         self.request.headers['Content-Length'] = MAX_REQUEST_BODY_SIZE + 1
46 +         self.request.body = "0" * (MAX_REQUEST_BODY_SIZE + 1)
47 +         self.assertRaises(exception.RequestTooLarge,
48 +                           self.request.get_response,
49 +                           self.middleware)
50 +
51 +     def test_request_too_large_no_content_length(self):
52 +         self.request.body = "0" * (MAX_REQUEST_BODY_SIZE + 1)
53 +         self.request.headers['Content-Length'] = None
54 +         self.assertRaises(exception.RequestTooLarge,
55 +                           self.request.get_response,
56 +                           self.middleware)
```

Comments 0

Please sign in to comment.