

Remove 3rd parameter from verify function

[Browse files](#)

master

parsonsbots committed 22 hours ago

1 parent [e284652](#)commit [e103dae92c32413bfafa62172570cfc65a74c4bd](#)

Showing 2 changed files with 31 additions and 30 deletions.

Unified

Split

23 checker.php

```
... @@ -1,16 +1,19 @@
1  <?php
2      class DedicatedIpChecker {
3
4      /**
5       * Get user ID for an IP address
6       *
7       * @param string $ip
8       * @param array $userIds
9       * - @param array $userIds
10      *
11      * @return string $response User ID for dedicated
12      IP (if user ID doesn't exist for IP, pick an ID from
13      existing user IDs)
14      */
15      - protected function _getUserId($ip,
16      $userIds, $userIds) {
17
18          if (!empty($userIds[$ip])) {
19              return $userIds[$ip];
20          }
21
22          $key = array_sum($subnets)
23
24      * count($subnets);
25
26      } else {
27          $subnets = explode('.',
28      $ip);
29
30      - $key = ($subnets[0] != 0 ?
31      $subnets[0] : 1) * ($subnets[3] != 0 ? $subnets[3] : 1) +
32      $subnets[2] + $subnets[1];
33
34      }
35
36      - $response = ($userIds[$key]);
37
38      return $response;
39
40      }
41
42      foreach ($splitIpSubnets
43
44      as $splitIpSubnet) {
45
46          if (
47
48      !is_numeric($splitIpSubnet) ||
49      - strlen($splitIpSubnet) > 3 ||
50
51      $splitIpSubnet > 255 ||
```

```
1  <?php
2      class DedicatedIpChecker {
3
4      + public function __construct() {
5      +     $this->userIds = array();
6      +     }
7      +
8      /**
9       * Get user ID for an IP address
10      *
11      * @param string $ip
12      * @param array $userIds
13      *
14      * @return string $response User ID for dedicated
15      IP (if user ID doesn't exist for IP, pick an ID from
16      existing user IDs)
17      */
18      + protected function _getUserId($ip,
19      $userIds) {
20
21          if (!empty($userIds[$ip])) {
22              return $userIds[$ip];
23          }
24
25          $key = array_sum($subnets)
26
27      * count($subnets);
28
29      } else {
30          $subnets = explode('.',
31      $ip);
32
33      + $key = max($subnets[0], 1)
34      * max($subnets[3], 1) + $subnets[2] + $subnets[1];
35
36      }
37
38      + $response = ($this->
39      userIds[$key]);
40
41      return $response;
42
43      }
44
45      foreach ($splitIpSubnets
46
47      as $splitIpSubnet) {
48
49          if (
50
51      !is_numeric($splitIpSubnet) ||
52      + strlen($splitIpSubnet) >= 4 ||
53
54      $splitIpSubnet > 255 ||
```

<pre> 72 \$splitIpSubnet < 0 73) { 152 */ 153 public function verify(\$ips, \$userIps) { 154 \$ips = \$this->_parseIps(\$ips); 155 - \$userIds = array_values(\$userIps); 156 157 if (!empty(\$ips)) { 158 \$userIdsFormatted = false; 159 160 do { 161 - \$userIds = 162 array_merge(\$userIds, \$userIps); 163 - 164 if 165 (count(\$userIds) > 255255) { 166 167 \$userIdsFormatted = true; 168 } 169 - } while (\$userIdsFormatted 170 == false); 171 172 \$ips = explode("\n", \$ips); 173 174 foreach (\$ips as \$key => \$ip) { 175 - \$ips[\$ip] = \$this- 176 >_getUserId(\$ip, \$userIps, \$userIds); 177 unset(\$ips[\$key]); 178 } </pre>	<pre> 75 \$splitIpSubnet < 0 76) { 155 */ 156 public function verify(\$ips, \$userIps) { 157 \$ips = \$this->_parseIps(\$ips); 158 + \$this->userIds = 159 array_values(\$userIps); 160 161 if (!empty(\$ips)) { 162 \$userIdsFormatted = false; 163 164 do { 165 + \$this->userIds = 166 array_merge(\$this->userIds, \$this->userIps); 167 + 168 if (count(\$this- 169 >userIds) > 255255) { 170 171 \$userIdsFormatted = true; 172 } 173 - } while (\$userIdsFormatted 174 == false); 175 176 \$ips = explode("\n", \$ips); 177 178 foreach (\$ips as \$key => \$ip) { </pre>
---	--

▼ 38 data.php

<pre> 6 * 7 */ 8 \$ipListToCheck = implode("\n", array(9 - 10 '2001:db8:abcd:0008:847g:3e2:1088:8888', 11 '2001:db8:abcd:0008:0000:0000:8888:ffff', 12 - 13 '2001:db8:abcd:0008::1234:8888:FFFF', 14 - 15 '172.16.88.10', 16 - 17 '172.16.8.3', 18 - 19 '172.16.200.3', 20 - 21 '172.16.201.4', 22 - 23 '192.168.0.0', 24 - 25 '192.168.0.1', 26 - 27 '192.168.0.2', 28 - 29 '192.168.0.3', 30 - 31 '192.168.0.4', 32 - 33 '192.168.40.4', 34 - 35 '192.168.49.4', 36 - 37 '10.5.30.103', 38 - 39 '10.5.30.105', 40 - 41 '10.5.30.106', 42 - 43 '10.5.31.108', 44 - 45 '_.10.5.30.199_*', // Invalid IP 46 formats will be parsed </pre>	<pre> 6 * 7 */ 8 \$ipListToCheck = implode("\n", array(9 + 10 '2001:db8:abcd:0008:847g:2e2:1088:8888', 11 '2001:db8:abcd:0008:0000:0000:8888:ffff', 12 + 13 '2001:db8:abcd:0008::1224:8888:FFFF', 14 + 15 '172.16.88.10', 16 + 17 '172.16.8.2', 18 + 19 '172.16.200.2', 20 + 21 '172.16.201.4', 22 + 23 '192.168.0.0', 24 + 25 '192.168.0.1', 26 + 27 '192.168.0.2', 28 + 29 '192.168.0.4', 30 + 31 '192.168.40.4', 32 + 33 '192.168.49.4', 34 + 35 '10.5.20.102', 36 + 37 '10.5.20.105', 38 + 39 '10.5.20.106', 40 + 41 '10.5.21.108', 42 + 43 '_.10.5.20.199_*', // Invalid IP 44 formats will be parsed </pre>
---	--

```

29         '10.5.89.44',
30         '10.8.8.100',
31         '10.8.8.101',
32         '10.8.8.102',
33 -       '10.8.8.103'
34     ));
35
36     /**
42     */
43     $userIps = array(
44
45         '2001:db8:abcd:0008:0000:0000:8888:ffff' => '5c9518c1-0ad8-
4e41-80a1-5bd54221ccee',
46 -       '192.168.0.2' => '5c95396e-97c8-
49af-abb3-17e04221ccee',
47 -       '192.168.0.3' => '5c95396e-97c8-
49af-abb3-17e04221ccee',
48 -       '10.5.31.108' => '5c95354f-43c8-
4976-b69d-12334221ccee',
49 -       '172.16.88.10' => '5c953514-e9e8-
4cca-954c-12334221ccee',
50 -       '172.16.200.3' => '5c953514-e9e8-
4cca-954c-12334221ccee',
51 -       '10.5.89.44' => '5c952170-f908-
45e9-af43-6bf64221ccee',
52 -       '10.5.30.199' => '5c9518c1-0ad8-
4e41-80a1-5bd54221ccee',
53 -       '172.16.8.3' => '5c951454-0418-
4ca5-80f8-5b6b4221ccee',
54 -       '192.168.40.4' => '5c951242-56a9-
401a-8d74-5b7b4221ccee',
55 -       '10.8.8.102' => '5c95396e-97c8-
49af-abb3-17e04221ccee'
56     );
?>

```

```

28         '10.5.89.44',
29         '10.8.8.100',
30         '10.8.8.101',
31         '10.8.8.102',
32
33     ));
34
35     /**
40     */
41     $userIps = array(
42
43         '2001:db8:abcd:0008:0000:0000:8888:ffff' => '5c9518c1-0ad8-
4e41-80a1-5bd54221ccee',
44 +       '192.168.0.2' => '5c95296e-97c8-
49af-abb2-17e04221ccee',
45 +       '192.168.2.0' => '5c95296e-97c8-
49af-abb2-17e04221ccee',
46 +       '10.5.21.108' => '5c95254f-42c8-
4976-b69d-11224221ccee',
47 +       '172.16.88.10' => '5c952514-e9e8-
4cca-954c-11224221ccee',
48 +       '172.16.200.2' => '5c952514-e9e8-
4cca-954c-11224221ccee',
49 +       '10.5.89.44' => '5c952170-f908-
45e9-af42-6bf64221ccee',
50 +       '10.5.20.199' => '5c9518c1-0ad8-
4e41-80a1-5bd54221ccee',
51 +       '172.16.8.2' => '5c951454-0418-
4ca5-80f8-5b6b4221ccee',
52 +       '192.168.40.4' => '5c951242-56a9-
401a-8d74-5b7b4221ccee',
53 +       '10.8.8.102' => '5c95296e-97c8-
49af-abb2-17e04221ccee'
54     );
?>

```

0 comments on commit `e103dae`