



Refactor proxy process port load balancing

[Browse files](#) master parsonsbots committed 27 seconds ago1 parent [0ee9acc](#)

commit 11175594983b5709a720c1bd3a2cd31792928b11

 Showing 1 changed file with 12 additions and 58 deletions.[Unified](#) [Split](#)▼ 70  dynamic_reconfiguration.php 

```
221      * @return array $firewallRules Firewall rules
222      */
223      protected function
      _applyFirewallRules($firewallPorts) {

224          $firewallRules = array(
225              '*filter',
226              ':INPUT ACCEPT [0:0]',
270              krsort($destinationIps);
271
272          foreach ($destinationIps
273 as $destinationIpKey => $destinationIp) {
274      -
275          $loadBalancer =
276      $destinationIpKey > 1 ? '-m statistic --mode nth --every '
277      . $destinationIpKey . ' --packet 0 ' : '';
278          $firewallRules[] =
279      '-A OUTPUT -d ' . $sourceIp . '/32 -p udp -m udp --dport
280      53 ' . $loadBalancer . '-j DNAT --to-destination ' .
281      $destinationIp;
282      }
283      }
284      // ..
285
286      /*
287
288      if (
289          !empty($this->processes)
290      &&
291          is_array($this->processes)
292      ) {
293          foreach ($this->processes
294 as $protocol => $processes) {
295              $basePorts =
296              array();
297
298              if ($redundant) {
299                  unset($processes[0]);
300              }
301          }
302      }
```

```
221      * @return array $firewallRules Firewall rules
222      */
223      protected function
      _applyFirewallRules($firewallPorts) {
224      +
225          if (empty($firewallPorts)) {
226              return false;
227          }
228
229          $firewallRules = array(
230              '*filter',
231              ':INPUT ACCEPT [0:0]',
274              krsort($destinationIps);
275
276          foreach ($destinationIps
277 as $destinationIpKey => $destinationIp) {
278      +
279          $loadBalancer =
280      $destinationIpKey > 0 ? '-m statistic --mode nth --every '
281      . ($destinationIpKey + 1) . ' --packet 0 ' : '';
282          $firewallRules[] =
283      '-A OUTPUT -d ' . $sourceIp . '/32 -p udp -m udp --dport
284      53 ' . $loadBalancer . '-j DNAT --to-destination ' .
285      $destinationIp;
286      }
287      }
288
289      foreach ($this->server['data']
290 ['forwarding_ports'] as $proxyProcessName =>
291 $proxyProcessForwardingPorts) {
292      +
293          krsort($firewallPorts[$proxyProcessName]);
294      +
295          $proxyProcessForwardingPorts = implode(',',
296          $proxyProcessForwardingPorts);
```

```

= array_values($processes);
292 -                                     }
293 -
294 -                                     if
295 - (!empty($overridePorts)) {
296 -                                     if
297 - (empty($overridePorts[$protocol])) {
298 -
299 -                                     $processes
300 -                                     }
301 -
302 -                                     foreach
303 - ($processes as $ports) {
304 -                                     $basePorts
305 - = array_merge($ports, $basePorts);
306 -                                     }
307 -                                     $ports =
308 - array_unique($basePorts);
309 -                                     foreach ($ports as
310 - $portKey => $port) {
311 -                                     if ($this->
312 - _checkPort(key($this->server['data']['proxy_ips']),
313 - $port, $protocol) === false) {
314 - unset($ports[$portKey]);
315 -                                     }
316 -                                     }
317 -                                     shuffle($ports);
318 -                                     $ports =
319 - array_values($ports);
320 -                                     krsort($ports);
321 -                                     $dports =
322 - array_merge($this->processes[$protocol][0], $basePorts);
323 -                                     $dportsSplit =
324 - array_chunk($dports, '5');
325 -                                     if
326 - (!empty($ports)) {
327 -                                     foreach
328 - ($dportsSplit as $chunk => $dports) {
329 -
330 - $dports = implode(',', $dports);
331 -
332 -                                     foreach ($ports as $portKey => $port) {
333 -
334 - $loadBalancer = '';

```

```

285 +
286 +                                     foreach
287 + ($firewallPorts[$proxyProcessName] as $proxyProcessPortKey
288 + => $proxyProcessPort) {
289 +
290 +                                     $loadBalancer =
291 + $proxyProcessPortKey > 0 ? '-m statistic --mode nth --
292 + every ' . ($proxyProcessPortKey + 1) . ' --packet 0 ' :
293 + '';
294 +
295 +                                     $firewallRules[] =
296 + '-A PREROUTING -p tcp -m multiport --dports ' .
297 + $proxyProcessForwardingPorts . ' ' . $loadBalancer . '-j
298 + DNAT --to-destination : ' . $proxyProcessPort . ' --
299 + persistent';

```

```

327 -
328 -
329 -
330 -
331 -
332 -
333 -
334 -
335 -
336 -
337 -
338 -
339 -

```

if (\$portKey > 0) {

\$loadBalancer = '-m statistic --mode nth --every ' .

(\$portKey + 1) . ' --packet 0 ';

}

\$firewallRules[] = '-A PREROUTING -p tcp -m multiport --

dports ' . \$dports . ' ' . \$loadBalancer . '-j DNAT --to-

destination :' . \$port . ' --persistent';

}

}

}

}/

\$firewallRules[] = 'COMMIT';

\$firewallRuleChunks =

array_chunk(\$firewallRules, 100);

```

289 +
290 +
291 +
292 +
293 +

```

}

}

\$firewallRules[] = 'COMMIT';

\$firewallRuleChunks =

array_chunk(\$firewallRules, 100);

0 comments on commit 1117559