parsonsbots / **dynamic-proxy-node-reconfiguration**

Refactor proxy process port load balancing

master

Browse files

parsonsbots committed 27 seconds ago

1 parent 0ee9acc commit 11175594983b5709a720c1bd3a2cd31792928b11

Showing 1 changed file with **12 additions** and **58 deletions**.

Unified Split

70 ■■■■□ dynamic_reconfiguration.php

```
        * @return array $firewallRules Firewall rules
        */
                protected function _applyFirewallRules($firewallPorts) {
                        $firewallRules = array(
                                '*filter',
                                ':INPUT ACCEPT [0:0]',
                                krsort($destinationIps);

                                foreach ($destinationIps as $destinationIpKey => $destinationIp) {
-                                       $loadBalancer = $destinationIpKey > 1 ? '-m statistic --mode nth --every ' . $destinationIpKey . ' --packet 0 ' : '';
                                        $firewallRules[] = '-A OUTPUT -d ' . $sourceIp . '/32 -p udp -m udp --dport 53 ' . $loadBalancer . '-j DNAT --to-destination ' . $destinationIp;
                                }
                        }

-                       // ..
-
-                       /*
-
-                       if (
-                               !empty($this->processes) &&
-                               is_array($this->processes)
-                       ) {
-                               foreach ($this->processes as $protocol => $processes) {
-                                       $basePorts = array();
-
-                                       if ($redundant) {
-                                               unset($processes[0]);
-                                                       $processes
```

```
        * @return array $firewallRules Firewall rules
        */
                protected function _applyFirewallRules($firewallPorts) {
+                       if (empty($firewallPorts)) {
+                               return false;
+                       }
+
                        $firewallRules = array(
                                '*filter',
                                ':INPUT ACCEPT [0:0]',
                                krsort($destinationIps);

                                foreach ($destinationIps as $destinationIpKey => $destinationIp) {
+                                       $loadBalancer = $destinationIpKey > 0 ? '-m statistic --mode nth --every ' . ($destinationIpKey + 1) . ' --packet 0 ' : '';
                                        $firewallRules[] = '-A OUTPUT -d ' . $sourceIp . '/32 -p udp -m udp --dport 53 ' . $loadBalancer . '-j DNAT --to-destination ' . $destinationIp;
                                }
                        }

+                       foreach ($this->server['data']['forwarding_ports'] as $proxyProcessName => $proxyProcessForwardingPorts) {
+                               krsort($firewallPorts[$proxyProcessName]);
+                               $proxyProcessForwardingPorts = implode(',', $proxyProcessForwardingPorts);
```

```
= array_values($processes);
-                                }
-
-                                if (!empty($overridePorts)) {
-                                        if (empty($overridePorts[$protocol])) {
-                                                continue;
-                                        }

-                                        $processes = $overridePorts[$protocol];
-                                }
-
-                                foreach ($processes as $ports) {
-                                        $basePorts = array_merge($ports, $basePorts);
-                                }
-
-                                $ports = array_unique($basePorts);
-
-                                foreach ($ports as $portKey => $port) {
-                                        if ($this->_checkPort(key($this->server['data']['proxy_ips']), $port, $protocol) === false) {
-                                                unset($ports[$portKey]);
-                                        }
-                                }
-
-                                shuffle($ports);
-                                $ports = array_values($ports);
-                                krsort($ports);
-                                $dports = array_merge($this->processes[$protocol][0], $basePorts);
-                                $dportsSplit = array_chunk($dports, '5');
-
-                                if (!empty($ports)) {
-                                        foreach ($dportsSplit as $chunk => $dports) {
-                                                $dports = implode(',', $dports);
-
-                                                foreach ($ports as $portKey => $port) {
-                                                        $loadBalancer = '';
-
```

```

+                                foreach ($firewallPorts[$proxyProcessName] as $proxyProcessPortKey => $proxyProcessPort) {
+                                        $loadBalancer = $proxyProcessPortKey > 0 ? '-m statistic --mode nth --every ' . ($proxyProcessPortKey + 1) . ' --packet 0 ' : '';
+                                        $firewallRules[] = '-A PREROUTING -p tcp -m multiport --dports ' . $proxyProcessForwardingPorts . ' ' . $loadBalancer . '-j DNAT --to-destination :' . $proxyProcessPort . ' --persistent';
-
if ($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';                                $firewallRules[] = 'COMMIT';
                        $firewallRuleChunks =                                       $firewallRuleChunks =
array_chunk($firewallRules, 100);                           array_chunk($firewallRules, 100);
```

0 comments on commit 1117559