

[Browse files](#)

🔗 master 🏷️ 1.0.0

```
0 parents    commit 5afea06f3bc31233d2970f7e03ef73626275ab35
```

Unified	Split
---------	-------

Large diffs are not rendered by default.

```
.. @@ -0,0 +1,14 @@
```

```
1 + SHELL=/bin/sh
2 + PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
3 +
4 + */10 * * * * php /scripts/reconfigure.php
5 + * * * * * php /scripts/firewall.php
6 + * * * * * sleep 10 && php /scripts/firewall.php
7 + * * * * * sleep 20 && php /scripts/firewall.php
8 + * * * * * sleep 30 && php /scripts/firewall.php
9 + * * * * * sleep 40 && php /scripts/firewall.php
10 + * * * * * sleep 50 && php /scripts/firewall.php
11 + * * * * * rm /var/log/squid3/*
12 + * * * * * rm /var/log/apache2/*
13 + * * * * * rm /var/log/3proxy/*
14 + * * * * * rm /var/cache/bind/named_dump.db
```

.. @@ -0,0 +1,103 @@

```

1 + <?php
2 +         // This file is part of the the temporary
   production configuration that isn't using standard OOP,
   DRY principles, etc.
3 +
4 +         date_default_timezone_set('America/Los_Angeles');
5 +         ini_set('memory_limit', -1);
6 +
7 +         if (
8 +
9 file_exists('/scripts/pid/reconfigure.pid') &&
10 +             empty($argv[1])
11 +         ) {
12 +             die();
13 +         }
14 +
15 +         $acls = explode("\n",

```

```

file_get_contents('/etc/squid3/proxy_ip_acl.conf'));
15 +     $gatewaysData =
json_decode(file_get_contents('/scripts/cache/gatewaysData
'), true);
16 +
17 +     if (
18 +         empty($acls) ||
19 +         empty($gatewaysData['data']['proxies'])
20 +     ) {
21 +         die();
22 +     }
23 +
24 +     $proxies = $gatewaysData['data']['proxies'];
25 +     $rules = array(
26 +         '*filter',
27 +         ':INPUT ACCEPT [0:0]',
28 +         ':FORWARD ACCEPT [0:0]',
29 +         ':OUTPUT ACCEPT [0:0]',
30 +         '-A INPUT -p icmp -m hashlimit --
hashlimit-name icmp --hashlimit-mode srcip --hashlimit
1/second --hashlimit-burst 2 -j ACCEPT',
31 +         '-A INPUT -p tcp -m tcp --dport 22 -m
connlimit --connlimit-above 4 --connlimit-mask 16 --
connlimit-saddr -j REJECT --reject-with tcp-reset',
32 +         '-A INPUT -p tcp -m tcp --dport 22 -m
hashlimit --hashlimit-upto 15/hour --hashlimit-burst 3 --
hashlimit-mode srcip --hashlimit-name ssh --hashlimit-
htable-expire 500000 -j ACCEPT',
33 +     );
34 +
35 +     if (
36 +         !empty($gatewaysData['data']
['firewall_filter']) &&
37 +         is_array($gatewaysData['data']
['firewall_filter'])
38 +     ) {
39 +         foreach ($gatewaysData['data']
['firewall_filter'] as $rule) {
40 +             $rules[] = $rule;
41 +         }
42 +     }
43 +
44 +     $rules[] = '-A OUTPUT -d 127.0.0.0/24 -p udp -m
udp -j ACCEPT';
45 +     $rules[] = '-A OUTPUT -s 127.0.0.0/24 -p udp -j
DROP';
46 +     $rules[] = 'COMMIT';
47 +     $rules[] = '*nat';
48 +     $rules[] = ':PREROUTING ACCEPT [0:0]';
49 +     $rules[] = ':INPUT ACCEPT [0:0]';
50 +     $rules[] = ':OUTPUT ACCEPT [0:0]';
51 +     $rules[] = ':POSTROUTING ACCEPT [0:0]';
52 +     $ports = array('80', '8888', '55555', '4444',
'7777', '55540', '55541', '55546', '55547', '55520',
'55521', '55524', '55525', '55526', '55527', '55528',
'55529', '55530', '55531', '55532', '55533', '55534',
'55535');
53 +
54 +     foreach ($ports as $key => $port) {
55 +         $response = shell_exec('curl -I -s -x ' .
$proxies[0] . ':' . $port . ' http://squid -v --connect-
timeout 2 --max-time 2');
56 +

```

```

57 +         if (
58 +             strpos(strtolower($response), '407
proxy') === false &&
59 +             strpos(strtolower($response), '403
forbidden') === false &&
60 +             strpos(strtolower($response), '503
') === false &&
61 +             strpos(strtolower($response), '
timed out ') === false
62 +         ) {
63 +             unset($ports[$key]);
64 +         }
65 +     }
66 +
67 +     shuffle($ports);
68 +     $ports = array_values($ports);
69 +     krsort($ports);
70 +     $dports = $ports;
71 +     $dportsSplit = array_chunk($dports, '5');
72 +
73 +     if (!empty($ports)) {
74 +         foreach ($dportsSplit as $chunk =>
$dports) {
75 +             $dports = implode(',', $dports);
76 +
77 +             foreach ($ports as $key => $port)
{
78 +                 $loadBalancer = '';
79 +
80 +                 if ($key > 0) {
81 +                     $loadBalancer = '-
m statistic --mode nth --every ' . ($key + 1) . ' --packet
0 ';
82 +                 }
83 +
84 +                 $rules[] = '-A PREROUTING
-p tcp -m multiport --dports ' . $dports . ' ' .
$loadBalancer . '-j DNAT --to-destination :' . $port . ' -
-persistent';
85 +             }
86 +         }
87 +     }
88 +
89 +     $rules[] = 'COMMIT';
90 +     $rules = array_chunk($rules, 100);
91 +     unlink('/scripts/iptables/iptables-elastic');
92 +     touch('/scripts/iptables/iptables-elastic');
93 +
94 +     foreach ($rules as $ruleChunk) {
95 +         $saveRules = implode("\n", $ruleChunk);
96 +         shell_exec('echo "' . $saveRules . '" >>
/scripts/iptables/iptables-elastic');
97 +     }
98 +
99 +     shell_exec('iptables-restore <
/scripts/iptables/iptables-elastic');
100 +     shell_exec('iptables-save');
101 +     gc_collect_cycles();
102 +     die();
103 + ?>

```

```
... @@ -0,0 +1,123 @@
```

```

1 + <?php
2 + // This file is part of the the temporary
  production configuration that isn't using standard OOP,
  DRY principles, etc.
3 +
4 +     date_default_timezone_set('America/Los_Angeles');
5 +     ini_set('memory_limit', -1);
6 +
7 +     if (empty($argv[1])) {
8 +         die();
9 +     }
10 +
11 +     $gatewaysData =
  json_decode(file_get_contents('/scripts/cache/gatewaysData
  '), true);
12 +     $proxies = $gatewaysData['data']['proxies'];
13 +     $rules = array(
14 +         '*filter',
15 +         ':INPUT ACCEPT [0:0]',
16 +         ':FORWARD ACCEPT [0:0]',
17 +         ':OUTPUT ACCEPT [0:0]',
18 +         '-A INPUT -p icmp -m hashlimit --
  hashlimit-name icmp --hashlimit-mode srcip --hashlimit
  1/second --hashlimit-burst 2 -j ACCEPT',
19 +         '-A INPUT -p tcp -m tcp --dport 22 -m
  connlimit --connlimit-above 4 --connlimit-mask 16 --
  connlimit-saddr -j REJECT --reject-with tcp-reset',
20 +         '-A INPUT -p tcp -m tcp --dport 22 -m
  hashlimit --hashlimit-upto 15/hour --hashlimit-burst 3 --
  hashlimit-mode srcip --hashlimit-name ssh --hashlimit-
  htable-expire 500000 -j ACCEPT',
21 +     );
22 +
23 +     if (
24 +         !empty($gatewaysData['data']
  ['firewall_filter']) &&
25 +         is_array($gatewaysData['data']
  ['firewall_filter'])
26 +     ) {
27 +         foreach ($gatewaysData['data']
  ['firewall_filter'] as $rule) {
28 +             $rules[] = $rule;
29 +         }
30 +     }
31 +
32 +     $rules[] = '-A OUTPUT -d 127.0.0.0/24 -p udp -m
  udp -j ACCEPT';
33 +     $rules[] = '-A OUTPUT -s 127.0.0.0/24 -p udp -j
  DROP';
34 +     $rules[] = 'COMMIT';
35 +     $rules[] = '*nat';
36 +     $rules[] = ':PREROUTING ACCEPT [0:0]';
37 +     $rules[] = ':INPUT ACCEPT [0:0]';
38 +     $rules[] = ':OUTPUT ACCEPT [0:0]';
39 +     $rules[] = ':POSTROUTING ACCEPT [0:0]';
40 +     $basePorts = array('80', '8888', '55555', '4444',
  '7777', '55540', '55541', '55546', '55547', '55520',
  '55521', '55524', '55525', '55526', '55527', '55528',
  '55529', '55530', '55531', '55532', '55533', '55534',
  '55535');
41 +     $ports = $basePorts;
42 +     $ports1 = array('4444', '7777');
```

```

43 +     $ports2 = array('55540', '55541');
44 +     $ports3 = array('55546', '55547');
45 +     $ports4 = array('55520', '55521');
46 +     $ports5 = array('55524', '55525');
47 +     $ports6 = array('55526', '55527');
48 +     $ports7 = array('55528', '55529');
49 +     $ports8 = array('55530', '55531');
50 +     $ports9 = array('55532', '55533');
51 +     $ports10 = array('55534', '55535');
52 +     $passedPorts = array_filter(array_map('trim',
explode('+', $argv[1])));
53 +     $activePorts = array(
54 +         '80',
55 +         '8888'
56 +     );
57 +
58 +     foreach ($passedPorts as $passedPort) {
59 +         if (!empty($passedPort)) {
60 +             foreach ($passedPort as $port) {
61 +                 if (in_array($port,
$ports)) {
62 +                                     $activePorts[] =
$port;
63 +                                     }
64 +             }
65 +         }
66 +     }
67 +
68 +     if (empty($activePorts)) {
69 +         die();
70 +     }
71 +
72 +     $ports = $activePorts;
73 +
74 +     foreach ($ports as $key => $port) {
75 +         $response = shell_exec('curl -I -s -x ' .
$proxies[0] . ':' . $port . ' http://squid -v --connect-
timeout 2 --max-time 2');
76 +
77 +         if (
78 +             strpos(strtolower($response), '407
proxy') === false &&
79 +             strpos(strtolower($response), '403
forbidden') === false &&
80 +             strpos(strtolower($response), '
503 ') === false &&
81 +             strpos(strtolower($response), '
timed out ') === false
82 +         ) {
83 +             unset($ports[$key]);
84 +         }
85 +     }
86 +
87 +     shuffle($ports);
88 +     $ports = array_values($ports);
89 +     krsort($ports);
90 +     $dports = $basePorts;
91 +     $dportsSplit = array_chunk($dports, '5');
92 +
93 +     if (!empty($ports)) {
94 +         foreach ($dportsSplit as $chunk =>
$dports) {
95 +             $dports = implode(',', $dports);

```

```

96 +
97 +             foreach ($ports as $key => $port)
98 +             {
99 +                 $loadBalancer = '';
100 +
101 +                 if ($key > 0) {
102 +                     $loadBalancer = '-
m statistic --mode nth --every ' . ($key + 1) . ' --packet
0 ' ;
103 +                 }
104 +
105 +                 $rules[] = '-A PREROUTING
-p tcp -m multiport --dports ' . $dports . ' ' .
$loadBalancer . '-j DNAT --to-destination :' . $port . ' -
-persistent';
106 +             }
107 +         }
108 +
109 +         $rules[] = 'COMMIT';
110 +         $rules = array_chunk($rules, 100);
111 +         unlink('/scripts/iptables/iptables-elastic');
112 +         touch('/scripts/iptables/iptables-elastic');
113 +
114 +         foreach ($rules as $ruleChunk) {
115 +             $saveRules = implode("\n", $ruleChunk);
116 +             shell_exec('echo "' . $saveRules . '" >>
/scripts/iptables/iptables-elastic');
117 +         }
118 +
119 +         shell_exec('iptables-restore <
/scripts/iptables/iptables-elastic');
120 +         shell_exec('iptables-save');
121 +         gc_collect_cycles();
122 +         die();
123 + ?>

```

▼ 380 ■■■■■ old_configuration_files/scripts/reconfigure.php 

... @@ -0,0 +1,380 @@

```

1 + <?php
2 +     // This file is part of the the temporary
production configuration that isn't using standard OOP,
DRY principles, etc.
3 +
4 +     date_default_timezone_set('America/Los_Angeles');
5 +     ini_set('memory_limit', -1);
6 +
7 +     if (!is_dir('/scripts/errors')) {
8 +         shell_exec('mkdir -m 777
/scripts/errors');
9 +     }
10 +
11 +     if (file_exists('/scripts/pid/reconfigure.pid')) {
12 +         $lastRan =
file_get_contents('/scripts/pid/reconfigure.pid');
13 +
14 +         if ($lastRan < strtotime('-15 minutes')) {
15 +
16 +             unlink('/scripts/pid/reconfigure.pid');
17 +         } else {
18 +             die();
19 +         }

```

```

19 +     }
20 +
21 +     $gatewaysJsonData = shell_exec("curl
https://ghostproxies.com/gateways/api.json --connect-
timeout 30");
22 +     $gatewaysData = json_decode($gatewaysJsonData,
true);
23 +     $ip = null;
24 +
25 +     if (
26 +         empty($gatewaysData['data']) ||
27 +         !is_dir('/etc/squid3')
28 +     ) {
29 +         file_put_contents('/scripts/errors/api-
error-' . time(), $gatewaysData);
30 +         unlink('/scripts/pid/reconfigure.pid');
31 +         die();
32 +     }
33 +
34 +     if (empty($gatewaysData['data']['squid_conf'])) {
35 +         unlink('/scripts/pid/reconfigure.pid');
36 +         die();
37 +     }
38 +
39 +     if (empty($gatewaysData['data']
['squid_redundant_conf'])) {
40 +         unlink('/scripts/pid/reconfigure.pid');
41 +         die();
42 +     }
43 +
44 +     if (empty($gatewaysData['data']['sysctl_conf'])) {
45 +         unlink('/scripts/pid/reconfigure.pid');
46 +         die();
47 +     }
48 +
49 +     if (!file_exists('/scripts/cache/gatewaysData')) {
50 +
51 +         file_put_contents('/scripts/cache/gatewaysData',
$gatewaysJsonData);
52 +     } else {
53 +         $previousData =
file_get_contents('/scripts/cache/gatewaysData');
54 +         file_put_contents('/scripts/cache/gatewaysData',
$gatewaysJsonData);
55 +     }
56 +     file_put_contents('/scripts/pid/reconfigure.pid',
time());
57 +     file_put_contents('/etc/squid3/squid.conf',
$gatewaysData['data']['squid_conf']);
58 +     file_put_contents('/etc/squid3/squid-
redundant.conf', $gatewaysData['data']
['squid_redundant_conf']);
59 +     file_put_contents('/etc/sysctl.conf',
$gatewaysData['data']['sysctl_conf']);
60 +     shell_exec('sysctl -p');
61 +
62 +     if (!is_dir('/scripts/cache')) {
63 +         shell_exec('mkdir -m 777 /scripts/cache');
64 +     }
65 +
66 +     foreach ($gatewaysData['data']['acls'] as $acl) {

```

```

67 +             if (strpos($acl, 'localip') !== false) {
68 +                 $splitAcl = explode(' ', $acl);
69 +                 $ip = $splitAcl[3];
70 +                 break;
71 +             }
72 +         }
73 +
74 +         if (empty($ip)) {
75 +             unlink('/scripts/pid/reconfigure.pid');
76 +             die();
77 +         }
78 +
79 +         shell_exec('rm -rf /etc/squid3/users/');
80 +         shell_exec('mkdir -m 777 /etc/squid3/users/');
81 +
82 +         if (!empty($gatewaysData['data']['files'])) {
83 +             foreach ($gatewaysData['data']['files'] as
84 $file) {
85 +                 shell_exec('mkdir -m 777 ' .
86 str_replace(array('s.txt', 'd.txt'), '', $file['path']));
87 +                 shell_exec('touch ' .
88 $file['path']);
89 +                 file_put_contents($file['path'],
90 $file['contents']);
91 +             }
92 +         }
93 +
94 +         $redundantAcls = array();
95 +         $proxies = $gatewaysData['data']['proxies'];
96 +         $rules = array(
97 +             '*filter',
98 +             ':INPUT ACCEPT [0:0]',
99 +             ':FORWARD ACCEPT [0:0]',
100 +             ':OUTPUT ACCEPT [0:0]',
101 +             '-A INPUT -p icmp -m hashlimit --
102 hashlimit-name icmp --hashlimit-mode srcip --hashlimit
103 1/second --hashlimit-burst 2 -j ACCEPT',
104 +             '-A INPUT -p tcp -m tcp --dport 22 -m
105 connlimit --connlimit-above 4 --connlimit-mask 16 --
106 connlimit-saddr -j REJECT --reject-with tcp-reset',
107 +             '-A INPUT -p tcp -m tcp --dport 22 -m
108 hashlimit --hashlimit-upto 15/hour --hashlimit-burst 3 --
109 hashlimit-mode srcip --hashlimit-name ssh --hashlimit-
110 htable-expire 500000 -j ACCEPT',
111 +         );
112 +
113 +         if (
114             !empty($gatewaysData['data']
115 ['firewall_filter']) &&
116             is_array($gatewaysData['data']
117 ['firewall_filter'])
118         ) {
119 +             foreach ($gatewaysData['data']
120 ['firewall_filter'] as $rule) {
121 +                 $rules[] = $rule;
122 +             }
123 +         }
124 +
125 +         $rules[] = '-A OUTPUT -d 127.0.0.0/24 -p udp -m
126 udp -j ACCEPT';
127 +         $rules[] = '-A OUTPUT -s 127.0.0.0/24 -p udp -j
128 DROP';
129 +         $rules[] = 'COMMIT';

```



```

114 +     $rules[] = '*nat';
115 +     $rules[] = ':PREROUTING ACCEPT [0:0]';
116 +     $rules[] = ':INPUT ACCEPT [0:0]';
117 +     $rules[] = ':OUTPUT ACCEPT [0:0]';
118 +     $rules[] = ':POSTROUTING ACCEPT [0:0]';
119 +     $basePorts = array('4444', '7777', '55540',
    '55541', '55546', '55547', '55520', '55521', '55524',
    '55525', '55526', '55527', '55528', '55529', '55530',
    '55531', '55532', '55533', '55534', '55535');
120 +     $ports = $basePorts;
121 +
122 +     foreach ($ports as $key => $port) {
123 +         $response = shell_exec('curl -I -s -x ' .
    $proxies[0] . ':' . $port . ' http://squid -v --connect-
    timeout 2 --max-time 2');
124 +
125 +         if (
126 +             strpos(strtolower($response), '407
    proxy') === false &&
127 +             strpos(strtolower($response), '403
    forbidden') === false &&
128 +             strpos(strtolower($response), '
    503 ') === false &&
129 +             strpos(strtolower($response), '
    timed out ') === false
130 +         ) {
131 +             unset($ports[$key]);
132 +         }
133 +     }
134 +
135 +     shuffle($ports);
136 +     $ports = array_values($ports);
137 +     krsort($ports);
138 +     $dports = array_merge(array('80', '8888',
    '55555'), $basePorts);
139 +     $dportsSplit = array_chunk($dports, '5');
140 +
141 +     if (!empty($ports)) {
142 +         foreach ($dportsSplit as $chunk =>
    $dports) {
143 +             $dports = implode(',', $dports);
144 +
145 +             foreach ($ports as $key => $port)
146 +             {
147 +                 $loadBalancer = '';
148 +
149 +                 if ($key > 0) {
150 +                     $loadBalancer = '-
    m statistic --mode nth --every ' . ($key + 1) . ' --packet
    0 ' ;
151 +                 }
152 +
153 +                 $rules[] = '-A PREROUTING
    -p tcp -m multiport --dports ' . $dports . ' ' .
    $loadBalancer . '-j DNAT --to-destination :' . $port . ' -
    -persistent';
154 +             }
155 +         }
156 +
157 +         $rules[] = '-A PREROUTING -p tcp --dport 1085 -j
    DNAT --to-destination :1090 --persistent';
158 +

```

```

159 +     shell_exec('rm /etc/squid3/proxy_ip_acl.conf');
160 +     shell_exec('touch /etc/squid3/proxy_ip_acl.conf');
161 +     file_put_contents('/etc/squid3/proxy_ip_acl.conf',
implode("\n", $gatewaysData['data']['acls']));
162 +     shell_exec('htpasswd -cb /etc/squid3/passwords
ghostproxies ghostproxies');
163 +     shell_exec('htpasswd -D /etc/squid3/passwords
ghostproxies');
164 +
165 +     if (!empty($gatewaysData['data']['users'])) {
166 +         foreach ($gatewaysData['data']['users'] as
$username => $password) {
167 +             shell_exec('htpasswd -b
/etc/squid3/passwords ' . $username . ' ' . $password);
168 +         }
169 +     }
170 +
171 +     exec('curl --socks5-hostname ' . $proxies[0] .
':1085' . ' http://socks/ -v --connect-timeout 4 --max-
time 4 2>&1', $socksResponse);
172 +     $socksResponse = end($socksResponse);
173 +     $socksAlive = (strpos(strtolower($socksResponse),
'empty reply ') !== false);
174 +     $rules[] = 'COMMIT';
175 +     $rules = array_chunk($rules, 100);
176 +     unlink('/scripts/iptables/iptables-redundant');
177 +     touch('/scripts/iptables/iptables-redundant');
178 +
179 +     foreach ($rules as $ruleChunk) {
180 +         $saveRules = implode("\n", $ruleChunk);
181 +         shell_exec('echo "' . $saveRules . '" >>
/scripts/iptables/iptables-redundant');
182 +     }
183 +
184 +     shell_exec('iptables-restore <
/scripts/iptables/iptables-redundant');
185 +     sleep(5);
186 +     shell_exec('iptables-save');
187 +
188 +     if (!empty($gatewaysData['data']['socks'])) {
189 +         sleep(10);
190 +
191 +         file_put_contents('/usr/local/etc/3proxy/3proxy.cfg',
$gatewaysData['data']['socks']);
192 +         $socksProcess = array_filter(explode("\n",
exec('ps -fC 3proxy.cfg')));
193 +
194 +         if (!empty($socksProcess[0])) {
195 +             $splitSocksProcess =
array_values(array_filter(explode(' ',
$socksProcess[0])));
196 +
197 +             if (
198 +
199 +                 !empty($splitSocksProcess[1]) &&
200 +                 is_numeric($splitSocksProcess[1])
201 +                 ) {
202 +                 $shell = '#!/bin/bash';
203 +                 $shell .= "\n" . 'kill ' .
($socksAlive ? '-SIGUSR1 ' : '') .
trim($splitSocksProcess[1]);

```

```

203     unlink('/scripts/socks.sh');
204     +
205     file_put_contents('/scripts/socks.sh', $shell);
206     +
207     shell_exec('chmod +x
208     /scripts/socks.sh');
209     +
210     shell_exec('/scripts/./socks.sh');
211     +
212     }
213     +
214     }
215     +
216     sleep(20);
217     +
218     $mainResponse = shell_exec('squid3 -k
219     reconfigure');
220     +
221     if (strpos(strtolower($mainResponse), ' error:')
222     !== false) {
223     +
224     exec('ps -fc squid3 2>&1', $processes);
225     +
226     if (!empty($processes[0])) {
227     +
228     unset($processes[0]);
229     +
230     foreach ($processes as $process) {
231     +
232     $processColumns =
233     array_values(array_filter(explode(' ', $process)));
234     +
235     if
236     (!empty($processColumns[1])) {
237     +
238     $killProcesses[] =
239     $processColumns[1];
240     +
241     }
242     +
243     }
244     +
245     if (!empty($killProcesses)) {
246     +
247     $shell = '#!/bin/bash';
248     +
249     foreach ($killProcesses as
250     $killProcess) {
251     +
252     $shell .= "\n" .
253     'kill ' . trim($killProcess);
254     +
255     }
256     +
257     unlink('/scripts/http.sh');
258     +
259     file_put_contents('/scripts/http.sh', $shell);
260     +
261     shell_exec('chmod +x
262     /scripts/http.sh');
263     +
264     shell_exec('/scripts/./http.sh');
265     +
266     }
267     +
268     }
269     +
270     unlink('/var/run/squid3.pid');
271     +
272     sleep(3);

```

```

249 +         shell_exec('squid3 start');
250 +     }
251 +
252 +     sleep(70);
253 +     unlink('/scripts/iptables/iptables-redundant');
254 +     shell_exec('php /scripts/firewall_redundant.php
ports6+ports7+ports8+ports9+ports10');
255 +
256 +     foreach (range(1, 5) as $instance) {
257 +         $instance = $instance == 1 ? '' :
$instance;
258 +         exec('squid3-redundant' . $instance . ' -k
reconfigure -f /etc/squid3/squid-redundant' . $instance .
'.conf 2>&1', $redundantResponse);
259 +         $redundantResponse =
array_shift($redundantResponse);
260 +
261 +         if (strpos(strtolower($redundantResponse),
' error:') !== false) {
262 +             exec('ps -fc squid3-redundant
2>&1', $processes);
263 +
264 +             if (!empty($processes[0])) {
265 +                 unset($processes[0]);
266 +
267 +                 foreach ($processes as
$process) {
268 +                     $processColumns =
array_values(array_filter(explode(' ', $process)));
269 +
270 +                     if (
271 +
!empty($processColumns[1]) &&
272 +
in_array('/etc/squid3/squid-redundant' . $instance .
'.conf', $processColumns)
273 +
) {
274 +
$killProcesses[] = $processColumns[1];
275 +
}
276 +
}
277 +
278 +             if
(!empty($killProcesses)) {
279 +
$shell =
'#!/bin/bash';
280 +
281 +             foreach
($killProcesses as $killProcess) {
282 +
$shell .=
"\n" . 'kill ' . trim($killProcess);
283 +
}
284 +
285 +
unlink('/scripts/http.sh');
286 +
file_put_contents('/scripts/http.sh', $shell);
287 +
shell_exec('chmod
+x /scripts/http.sh');
288 +
shell_exec('/scripts/./http.sh');
289 +
}
290 +
}

```

```

291 +
292 +             shell_exec('rm /var/run/squid-
redundant' . $instance . '.pid');
293 +             sleep(3);
294 +             shell_exec('squid3-redundant' .
$instance . ' start -f /etc/squid3/squid-redundant' .
$instance . '.conf');
295 +         }
296 +     }
297 +
298 +     sleep(70);
299 +     shell_exec('php /scripts/firewall_redundant.php
ports1+ports2+ports3+ports4+ports5');
300 +
301 +     foreach (range(6, 10) as $instance) {
302 +         exec('squid3-redundant' . $instance . ' -k
reconfigure -f /etc/squid3/squid-redundant' . $instance .
'.conf 2>&1', $redundantResponse);
303 +         $redundantResponse =
array_shift($redundantResponse);
304 +
305 +         if (strpos(strtolower($redundantResponse),
' error:') !== false) {
306 +             exec('ps -fc squid3-redundant
2>&1', $processes);
307 +
308 +             if (!empty($processes[0])) {
309 +                 unset($processes[0]);
310 +
311 +                 foreach ($processes as
$process) {
312 +                     $processColumns =
array_values(array_filter(explode(' ', $process)));
313 +
314 +                     if (
315 +                         !empty($processColumns[1]) &&
316 +                         in_array('/etc/squid3/squid-redundant' . $instance .
'.conf', $processColumns)
317 +                     ) {
318 +                         $killProcesses[] = $processColumns[1];
319 +                     }
320 +                 }
321 +
322 +                 if
(!empty($killProcesses)) {
323 +                     $shell =
'#!/bin/bash';
324 +
325 +                     foreach
($killProcesses as $killProcess) {
326 +                         $shell .=
"\n" . 'kill ' . trim($killProcess);
327 +                     }
328 +
329 +                     unlink('/scripts/http.sh');
330 +
331 +                     file_put_contents('/scripts/http.sh', $shell);
331 +                     shell_exec('chmod
+x /scripts/http.sh');

```

```

332 +
333 +         shell_exec('/scripts/./http.sh');
334 +     }
335 +
336 +         shell_exec('rm /var/run/squid-
redundant' . $instance . '.pid');
337 +         sleep(3);
338 +         shell_exec('squid3-redundant' .
$instance . ' start -f /etc/squid3/squid-redundant' .
$instance . '.conf');
339 +     }
340 + }
341 +
342 +     sleep(70);
343 +     shell_exec('php /scripts/firewall.php override');
344 +
345 +     if (!empty($gatewaysData['data']['socks-
redundant'])) {
346 +         sleep(15);
347 +         shell_exec('php /scripts/firewall.php');
348 +         exec('curl --socks5-hostname ' .
$proxies[0] . ':1090' . ' http://socks/ -v --connect-
timeout 4 --max-time 4 2>&1', $socksResponseRedundant);
349 +         $socksResponseRedundant =
end($socksResponseRedundant);
350 +         $socksAliveRedundant =
(strpos(strtolower($socksResponseRedundant), 'empty reply
') !== false);
351 +
352 +         file_put_contents('/usr/local/etc/3proxy/3proxy-
redundant.cfg', $gatewaysData['data']['socks-redundant']);
353 +         $socksProcessRedundant =
array_filter(explode("\n", exec('ps -fc 3proxy-
redundant.cfg')));
354 +         if (!empty($socksProcessRedundant[0])) {
355 +             $splitSocksProcessRedundant =
array_values(array_filter(explode(' ',
$socksProcessRedundant[0])));
356 +
357 +             if (
358 +
359 +                 !empty($splitSocksProcessRedundant[1]) &&
360 +                 is_numeric($splitSocksProcessRedundant[1])
361 +             ) {
362 +                 $shell = '#!/bin/bash';
363 +                 $shell .= "\n" . 'kill ' .
($socksAliveRedundant ? '-SIGUSR1 ' : '') .
trim($splitSocksProcessRedundant[1]);
364 +
365 +                 unlink('/scripts/socks.sh');
366 +
367 +                 file_put_contents('/scripts/socks.sh', $shell);
368 +                 shell_exec('chmod +x
/scripts/socks.sh');
369 +                 shell_exec('/scripts/./socks.sh');
370 +             }
371 +
372 +             if (!$socksAliveRedundant) {

```

11/23/2019

Initial commit · parsonsbots/dynamic-proxy-node-reconfiguration@5afea06

```

371 +                               sleep(10);
372 +                               shell_exec('service
373 +                               3proxy-redundant start');
374 +                               }
375 +                               }
376 +
377 +                               shell_exec('php /scripts/firewall.php override');
378 +                               unlink('/scripts/pid/reconfigure.pid');
379 +                               die();
380 + ?>

```

0 comments on commit 5afea06