


Update dynamic_reconfiguration.php

[Browse files](#)

master 1.0.0

 parsonsbots committed on Mar 26



1 parent 2e94435

commit d09e8c2c297e68c4cff0970c6debb2a2dd5f7189

Showing 1 changed file with 246 additions and 285 deletions.

Unified

Split

531  dynamic_reconfiguration.php 

```
13
14     public $api;
15     public $processes;
16
17     public $sshPorts;
18
19     public function __construct($api, $processes,
20 $protocols) {
21         $this->api = $api;
22         $this->processes = $processes;
23
24         $this->sshPorts = $sshPorts;
25     }
26 }
```

```
24 /**
25  * Configure firewall rules
26  *
27  * @param array $gatewaysData Data from gateway API
28  * response
29  */
30 public function configureFirewallRules($gatewaysData, $ruleSet) {
31     // Unlink old iptables rules
32     unlink('/scripts/iptables/iptables-' . $ruleSet);
33
34     // Create new iptables rules
35     touch('/scripts/iptables/iptables-' . $ruleSet);
36
37     // Apply rules
38     shell_exec('iptables-restore < /scripts/iptables/iptables-' . $ruleSet);
39
40     // Sleep for 5 seconds
41     sleep(5);
42
43     // Save rules
44     shell_exec('iptables-save');
45
46     return;
47 }
48
49 /**
50  * Configure gateway API
51  *
52  * @param array $gatewaysData Data from gateway API
53  * response
54  */
55 public function configureGatewayAPI($gatewaysData) {
56     // Parse gateway API response
57     $gatewaysData = json_decode($gatewaysData, true);
58
59     // Create gateway API response
60     $this->gatewaysData = $gatewaysData;
61 }
```

```
13
14     public $api;
15     public $processes;
16     + public $shell;
17     public $sshPorts;
18
19     public function __construct($api, $processes,
20 $protocols) {
21         $this->api = $api;
22         $this->processes = $processes;
23     + $this->shell = $shell;
24         $this->sshPorts = $sshPorts;
25     }
26
27 + /**
28 +  * Apply firewall rules
29 +  *
30 +  * @param array $firewallRules Firewall rules
31 +  * @param string $ruleSet Firewall rule set
32 +  *
33 +  * @return
34 +  */
35 + protected function
36 _applyFirewallRules($firewallRules, $ruleSet) {
37     + unlink('/scripts/iptables/iptables-' .
38 ruleSet);
39     + touch('/scripts/iptables/iptables-' .
40 ruleSet);
41
42     + foreach ($firewallRules as $ruleChunk) {
43         + $saveRules = implode("\n",
44 $ruleChunk);
45         + shell_exec('echo "' . $saveRules .
46 '" >> /scripts/iptables/iptables-' . ruleSet);
47     }
48
49     + shell_exec('iptables-restore <
50 /scripts/iptables/iptables-' . ruleSet);
51     + sleep(5);
52     + shell_exec('iptables-save');
53     + return;
54 }
55
56 + /**
57 +  * Configure firewall rules
58 +  *
59 +  * @param array $gatewaysData Data from gateway API
60 +  * response
61 +  */
62 public function configureGatewayAPI($gatewaysData) {
63     // Parse gateway API response
64     $gatewaysData = json_decode($gatewaysData, true);
65
66     // Create gateway API response
67     $this->gatewaysData = $gatewaysData;
68 }
```

```

28      *
29 - * @return $rules
30      */
31 -     protected function
_configureFirewallRules($gatewaysData) {
32
33         // Begin input filter rules
34         $rules = array(
35             '*filter',
36             is_array($this->processes)
37         ) {
38             foreach ($this->processes as
$protocol => $processes) {
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102

```

```

53 + * @param boolean $redundant Route to redundant process
ports only if true, all ports if false
54 + * @param array $ports Override default base ports
55      *
56 + * @return array $rules Firewall rules
57      */
58 +     protected function
_configureFirewallRules($gatewaysData, $redundant = false,
$overridePorts = array()) {
59
60         // Begin input filter rules
61         $rules = array(
62             '*filter',
63             is_array($this->processes)
64         ) {
65             foreach ($this->processes as
$protocol => $processes) {
66
67                 + if ($redundant) {
68                 +
69                 unset($processes[0]);
70                 +
71                 }
72                 +
73                 foreach ($processes as
$ports) {
74
75                     $basePorts =
76                     array_merge($ports, $basePorts);
77                     }
78                     $ports =
79                     array_unique($basePorts);
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142

```

```

103         shuffle($ports);
104         $ports =
105         array_values($ports);
106         krsort($ports);
107     */
108     protected function _createDirectories() {
109         $directories = array(
110
111             '/scripts/cache',
112 -            '/scripts/errors'
113
114         );
115
116         foreach ($directories as $directory) {
117
118         }
119
120     /**
121 -    * Reconfigure process
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195     *
196     * @param string $protocol Proxy protocol (http or
197     socks)

```

```

131 simple, persistent internal load balancing
143 +                                     method for DNAT
144 ports only (without changing the destination IP). This
145 also prevents malicious processes from hijacking
146 +                                     shared ports (when
147 the combination of SO_REUSEPORT and SO_REUSEADDR are used)
148 while still allowing a large number of sockets
149 +                                     */
150                                     shuffle($ports);
151                                     $ports =
152 array_values($ports);
153                                     krsort($ports);
154
155 */
156
157 protected function _createDirectories() {
158     $directories = array(
159 +         '/scripts',
160 +         '/scripts/cache',
161 +         '/scripts/errors',
162 +         '/scripts/iptables',
163 +         '/scripts/pid'
164     );
165
166     foreach ($directories as $directory) {
167
168     }
169
170 /**
171 + * Get process IDs from process name
172 + *
173 + * @param string $processName Process name
174 + *
175 + * @return array $processIds Process IDs
176 + */
177
178 protected function _getProcessIds($processName) {
179 +     $processIds = array();
180 +     exec('ps -fc ' . $processName . ' 2>&1',
181 $processes);
182
183 +
184 +     if (!empty($processes[0])) {
185 +         unset($processes[0]);
186 +
187 +         foreach ($processes as $process) {
188 +             $processColumns =
189 array_values(array_filter(explode(' ', $process)));
190 +
191 +             if
192 (!empty($processColumns[1])) {
193 +                 $processIds[] =
194 $processColumns[1];
195 +             }
196 +         }
197 +     }
198
199     return $processIds;
200 }
201
202 /**
203 + * Reconfigure specific process
204 + *
205 + * @param string $protocol Proxy protocol (http or
206 socks)
207 + *
208 + * @param string $binary Name of system binary
209 + *
210 + * @param string $command Software-specific command for
211 reconfiguration

```

```

197 * @param string $processName Process name
198 * @param string $processId Full path to process ID
199 - * @param string $aclFile Full path to ACL filename
200 - * @param string $aclData ACL data for $aclFile
201 * @param integer $delayStart Delay at beginning of
reconfiguration
202 * @param integer $delayEnd Delay at end of
reconfiguration
203 *
204 * @return
205 */
206 - protected function _reconfigure($protocol,
$processName, $processId, $aclFile, $aclData, $delayStart
= 0, $delayEnd = 0) {
207     if ($delayStart) {
208         sleep($delayStart);
209     }
210
211 - // Reconfigure processes here

```

```

271 + * @param string $errorString Reconfiguration error
string for error detection
272 * @param string $processName Process name
273 * @param string $processId Full path to process ID
274 * @param integer $delayStart Delay at beginning of
reconfiguration
275 * @param integer $delayEnd Delay at end of
reconfiguration
276 *
277 * @return
278 */
279 + protected function _reconfigure($protocol,
$binary, $command, $errorString, $processName, $processId,
$delayStart = 0, $delayEnd = 0) {
280     if ($delayStart) {
281         sleep($delayStart);
282     }
283
284 + $killProcesses = array();
285 + $shellCommands = array(
286 +     '#!' . $this->shell
287 + );
288 +
289 + if (!empty($command)) {
290 +     exec($command . ' 2>&1',
$reconfigureResponse);
291 +     $reconfigureResponse =
array_shift($reconfigureResponse);
292 + }
293 +
294 + if (
295 +     (
296 +     !empty($reconfigureResponse) &&
297 +         !empty($errorString) &&
298 +         strpos(strtolower($reconfigureResponse), $errorString) !==
false
299 +     ) ||
300 +     empty($command)
301 + ) {
302 +     $killProcesses = $this->
getProcessIds($processName);
303 + }
304 +
305 + foreach ($killProcesses as $killProcess) {
306 +     $shellCommands[] = 'kill ' .
trim($killProcess);
307 + }
308 +
309 + // Run shell commands through bash script
310 + if (count($shellCommands > 1)) {
311 +     unlink('/scripts/' . $protocol .
'.sh');
312 +     file_put_contents('/scripts/' .
$protocol . '.sh', implode("\n", $shellCommands));
313 +     shell_exec('chmod +x /scripts/' .
$protocol . '.sh');
314 +     shell_exec('/scripts/./' .
$protocol . '.sh');
315 + }

```

```

212
213         if ($delayEnd) {
214             sleep($delayEnd);
215         }
216     }
217
218     /**
219     - * Reconfigure redundant processes
220     *
221     - * @return boolean $status
222
223     */
224     protected function _applyReconfiguration() {
225         // Create writable log and cache
226         directories
227         // Apply primary Squid config
228
229         file_put_contents('/etc/squid3/squid.conf',
230             $gatewaysData['data']['squid_conf']);
231
232         // TODO: Apply redundant squid
233         configuration to all redundant processes
234         file_put_contents('/etc/squid3/squid-
235             redundant.conf', $gatewaysData['data']
236             ['squid_redundant_conf']);
237
238         // Save and apply sysctl settings
239     }
240
241     // Process recovery
242     unlink($processId);
243     sleep(5);
244     shell_exec('service ' . $binary .
245         ' start');
246 }
247
248     if ($delayEnd) {
249         sleep($delayEnd);
250     }
251 }
252
253 /**
254 + * Apply firewall
255 *
256 + * @return boolean
257 + */
258 + protected function _applyFirewall() {
259 +     if
260 +     (file_exists('/scripts/pid/reconfigure.pid')) {
261 +         return false;
262 +     }
263 +
264 +     $gatewaysData =
265 +     json_decode(file_get_contents('/scripts/cache/gatewaysData
266 +     '), true);
267 +
268 +     if (empty($gatewaysData['data']
269 +     ['proxies'])) {
270 +         return false;
271 +     }
272 +
273 +     $overridePorts = array(
274 +         'http' => $this->processes['http']
275 +     );
276 +     $firewallRules = $this-
277 +     >_configureFirewallRules($gatewaysData, false,
278 +     $overridePorts);
279 +     $this->_applyFirewallRules($firewallRules,
280 +     'elastic');
281 + }
282
283 + /**
284 + * Apply seamless processes reconfiguration
285 + *
286 + * @return boolean
287 + */
288 + protected function _applyReconfiguration() {
289 +     // Create writable log and cache
290 +     directories
291 +     // Apply primary Squid config
292
293 +     file_put_contents('/etc/squid3/squid.conf',
294 +         $gatewaysData['data']['squid_conf']);
295 +
296 +     // TODO: Apply global redundant squid
297 +     configuration to all redundant processes
298 +     file_put_contents('/etc/squid3/squid-
299 +         redundant.conf', $gatewaysData['data']
300 +         ['squid_redundant_conf']);
301 +
302 +     // Save and apply sysctl settings
303 + }

```

```

296         }
297
298     -         // Retrieve firewall rules
299     -         $firewallRules = $this->
300     >_configureFirewallRules($gatewaysData);
301
302         // Save Squid ACLs from API to file
303         shell_exec('rm
304         /etc/squid3/proxy_ip_acl.conf');
305         shell_exec('touch
306         /etc/squid3/proxy_ip_acl.conf');
307
308         file_put_contents('/etc/squid3/proxy_ip_acl.conf',
309         implode("\n", $gatewaysData['data']['acls']));
310
311     -         // Set proxy usernames and passwords using
312     -         htpasswd and basic_ncsa_auth
313     -         shell_exec('htpasswd -cb
314     -         /etc/squid3/passwords default default');
315     -         shell_exec('htpasswd -D
316     -         /etc/squid3/passwords default');
317
318         }
319
320         // Apply redundant firewall rules to begin
321         seamless reconfiguration
322     -         unlink('/scripts/iptables/iptables-
323     -         redundant');
324     -         touch('/scripts/iptables/iptables-
325     -         redundant');
326
327         foreach ($rules as $ruleChunk) {
328
329             $saveRules = implode("\n",
330             $ruleChunk);
331
332             shell_exec('echo "' . $saveRules .
333             '" >> /scripts/iptables/iptables-redundant');
334
335             }
336
337             shell_exec('iptables-restore <
338             /scripts/iptables/iptables-redundant');
339
340             sleep(5);
341
342             shell_exec('iptables-save');
343
344             // Reconfigure existing SOCKS process
345             first once redundant firewall is applied
346
347             if (
348                 !empty($gatewaysData['data']
349                 ['socks']) &&
350                 !empty($this->processes['socks']
351                 [0])
352             ) {
353
354                 // Check for alive primary SOCKS 5
355                 port
356
357                 $socksAlive = false;
358
359                 foreach ($this->processes['socks']
360                 [0] as $socksPort) {
361
362                     if (($socksAlive = $this->
363                     >_checkPort($proxies[0], $socksPort, 'socks')) === true) {
364
365                         break;
366
367                     }
368
369                 }
370
371                 // Reconfigure existing SOCKS processes
372                 first once redundant firewall is applied
373
374                 if (
375                     !empty($gatewaysData['data']
376                     ['socks']) &&
377                     !empty($this->processes['socks']
378                     [0])
379                 ) {
380
381                     // SOCKS config and ACLs are in
382                     the same file (TODO: move squid configuration and ACLs to
383                     same file for consistency)
384
385                     $this->_configureFirewallRules($gatewaysData, true);
386
387                     $this->_applyFirewallRules($firewallRules,
388                     'redundant');
389
390                     // Reconfigure existing SOCKS processes
391                     first once redundant firewall is applied
392
393                     $this->_reconfigureSOCKS($proxies, $socksPort, 'socks');
394
395                 }
396
397             }
398
399         }
400
401     }
402
403     }
404
405     }
406
407     }
408
409     }
410
411     }
412
413     }
414
415     }
416
417     }
418
419     }
420
421     }
422
423     }
424
425     }
426
427     }
428
429     }
430
431     }
432
433     }
434
435     }
436
437     }
438
439     }
440
441     }
442
443     }
444
445     }
446
447     }
448
449     }
450
451     }
452
453     }
454
455     }
456
457     }
458
459     }
460
461     }
462
463     }
464
465     }
466
467     }
468
469     }
470
471     }
472
473     }
474
475     }
476
477     }
478
479     }
480
481     }
482
483     }
484
485     }
486
487     }
488
489     }
490
491     }
492
493     }
494
495     }
496
497     }
498
499     }
500
501     }
502
503     }
504
505     }
506
507     }
508
509     }
510
511     }
512
513     }
514
515     }
516
517     }
518
519     }
520
521     }
522
523     }
524
525     }
526
527     }
528
529     }
530
531     }
532
533     }
534
535     }
536
537     }
538
539     }
540
541     }
542
543     }
544
545     }
546
547     }
548
549     }
550
551     }
552
553     }
554
555     }
556
557     }
558
559     }
560
561     }
562
563     }
564
565     }
566
567     }
568
569     }
570
571     }
572
573     }
574
575     }
576
577     }
578
579     }
580
581     }
582
583     }
584
585     }
586
587     }
588
589     }
590
591     }
592
593     }
594
595     }
596
597     }
598
599     }
600
601     }
602
603     }
604
605     }
606
607     }
608
609     }
610
611     }
612
613     }
614
615     }
616
617     }
618
619     }
620
621     }
622
623     }
624
625     }
626
627     }
628
629     }
630
631     }
632
633     }
634
635     }
636
637     }
638
639     }
640
641     }
642
643     }
644
645     }
646
647     }
648
649     }
650
651     }
652
653     }
654
655     }
656
657     }
658
659     }
660
661     }
662
663     }
664
665     }
666
667     }
668
669     }
670
671     }
672
673     }
674
675     }
676
677     }
678
679     }
680
681     }
682
683     }
684
685     }
686
687     }
688
689     }
690
691     }
692
693     }
694
695     }
696
697     }
698
699     }
700
701     }
702
703     }
704
705     }
706
707     }
708
709     }
710
711     }
712
713     }
714
715     }
716
717     }
718
719     }
720
721     }
722
723     }
724
725     }
726
727     }
728
729     }
730
731     }
732
733     }
734
735     }
736
737     }
738
739     }
740
741     }
742
743     }
744
745     }
746
747     }
748
749     }
750
751     }
752
753     }
754
755     }
756
757     }
758
759     }
760
761     }
762
763     }
764
765     }
766
767     }
768
769     }
770
771     }
772
773     }
774
775     }
776
777     }
778
779     }
780
781     }
782
783     }
784
785     }
786
787     }
788
789     }
790
791     }
792
793     }
794
795     }
796
797     }
798
799     }
800
801     }
802
803     }
804
805     }
806
807     }
808
809     }
810
811     }
812
813     }
814
815     }
816
817     }
818
819     }
820
821     }
822
823     }
824
825     }
826
827     }
828
829     }
830
831     }
832
833     }
834
835     }
836
837     }
838
839     }
840
841     }
842
843     }
844
845     }
846
847     }
848
849     }
850
851     }
852
853     }
854
855     }
856
857     }
858
859     }
860
861     }
862
863     }
864
865     }
866
867     }
868
869     }
870
871     }
872
873     }
874
875     }
876
877     }
878
879     }
880
881     }
882
883     }
884
885     }
886
887     }
888
889     }
890
891     }
892
893     }
894
895     }
896
897     }
898
899     }
900
901     }
902
903     }
904
905     }
906
907     }
908
909     }
910
911     }
912
913     }
914
915     }
916
917     }
918
919     }
920
921     }
922
923     }
924
925     }
926
927     }
928
929     }
930
931     }
932
933     }
934
935     }
936
937     }
938
939     }
940
941     }
942
943     }
944
945     }
946
947     }
948
949     }
950
951     }
952
953     }
954
955     }
956
957     }
958
959     }
960
961     }
962
963     }
964
965     }
966
967     }
968
969     }
970
971     }
972
973     }
974
975     }
976
977     }
978
979     }
980
981     }
982
983     }
984
985     }
986
987     }
988
989     }
990
991     }
992
993     }
994
995     }
996
997     }
998
999     }
1000
1001     }
1002
1003     }
1004
1005     }
1006
1007     }
1008
1009     }
1010
1011     }
1012
1013     }
1014
1015     }
1016
1017     }
1018
1019     }
1020
1021     }
1022
1023     }
1024
1025     }
1026
1027     }
1028
1029     }
1030
1031     }
1032
1033     }
1034
1035     }
1036
1037     }
1038
1039     }
1040
1041     }
1042
1043     }
1044
1045     }
1046
1047     }
1048
1049     }
1050
1051     }
1052
1053     }
1054
1055     }
1056
1057     }
1058
1059     }
1060
1061     }
1062
1063     }
1064
1065     }
1066
1067     }
1068
1069     }
1070
1071     }
1072
1073     }
1074
1075     }
1076
1077     }
1078
1079     }
1080
1081     }
1082
1083     }
1084
1085     }
1086
1087     }
1088
1089     }
1090
1091     }
1092
1093     }
1094
1095     }
1096
1097     }
1098
1099     }
1100
1101     }
1102
1103     }
1104
1105     }
1106
1107     }
1108
1109     }
1110
1111     }
1112
1113     }
1114
1115     }
1116
1117     }
1118
1119     }
1120
1121     }
1122
1123     }
1124
1125     }
1126
1127     }
1128
1129     }
1130
1131     }
1132
1133     }
1134
1135     }
1136
1137     }
1138
1139     }
1140
1141     }
1142
1143     }
1144
1145     }
1146
1147     }
1148
1149     }
1150
1151     }
1152
1153     }
1154
1155     }
1156
1157     }
1158
1159     }
1160
1161     }
1162
1163     }
1164
1165     }
1166
1167     }
1168
1169     }
1170
1171     }
1172
1173     }
1174
1175     }
1176
1177     }
1178
1179     }
1180
1181     }
1182
1183     }
1184
1185     }
1186
1187     }
1188
1189     }
1190
1191     }
1192
1193     }
1194
1195     }
1196
1197     }
1198
1199     }
1200
1201     }
1202
1203     }
1204
1205     }
1206
1207     }
1208
1209     }
1210
1211     }
1212
1213     }
1214
1215     }
1216
1217     }
1218
1219     }
1220
1221     }
1222
1223     }
1224
1225     }
1226
1227     }
1228
1229     }
1230
1231     }
1232
1233     }
1234
1235     }
1236
1237     }
1238
1239     }
1240
1241     }
1242
1243     }
1244
1245     }
1246
1247     }
1248
1249     }
1250
1251     }
1252
1253     }
1254
1255     }
1256
1257     }
1258
1259     }
1260
1261     }
1262
1263     }
1264
1265     }
1266
1267     }
1268
1269     }
1270
1271     }
1272
1273     }
1274
1275     }
1276
1277     }
1278
1279     }
1280
1281     }
1282
1283     }
1284
1285     }
1286
1287     }
1288
1289     }
1290
1291     }
1292
1293     }
1294
1295     }
1296
1297     }
1298
1299     }
1300
1301     }
1302
1303     }
1304
1305     }
1306
1307     }
1308
1309     }
1310
1311     }
1312
1313     }
1314
1315     }
1316
1317     }
1318
1319     }
1320
1321     }
1322
1323     }
1324
1325     }
1326
1327     }
1328
1329     }
1330
1331     }
1332
1333     }
1334
1335     }
1336
1337     }
1338
1339     }
1340
1341     }
1342
1343     }
1344
1345     }
1346
1347     }
1348
1349     }
1350
1351     }
1352
1353     }
1354
1355     }
1356
1357     }
1358
1359     }
1360
1361     }
1362
1363     }
1364
1365     }
1366
1367     }
1368
1369     }
1370
1371     }
1372
1373     }
1374
1375     }
1376
1377     }
1378
1379     }
1380
1381     }
1382
1383     }
1384
1385     }
1386
1387     }
1388
1389     }
1390
1391     }
1392
1393     }
1394
1395     }
1396
1397     }
1398
1399     }
1400
1401     }
1402
1403     }
1404
1405     }
1406
1407     }
1408
1409     }
1410
1411     }
1412
1413     }
1414
1415     }
1416
1417     }
1418
1419     }
1420
1421     }
1422
1423     }
1424
1425     }
1426
1427     }
1428
1429     }
1429
1430     }
1431
1432     }
1433
1434     }
1435
1436     }
1437
1438     }
1439
1440     }
1440
1441     }
1441
1442     }
1442
1443     }
1443
1444     }
1444
1445     }
1445
1446     }
1446
1447     }
1447
1448     }
1448
1449     }
1449
1450     }
1450
1451     }
1451
1452     }
1452
1453     }
1453
1454     }
1454
1455     }
1455
1456     }
1456
1457     }
1457
1458     }
1458
1459     }
1459
1460     }
1460
1461     }
1461
1462     }
1462
1463     }
1463
1464     }
1464
1465     }
1465
1466     }
1466
1467     }
1467
1468     }
1468
1469     }
1469
1470     }
1470
1471     }
1471
1472     }
1472
1473     }
1473
1474     }
1474
1475     }
1475
1476     }
1476
1477     }
1477
1478     }
1478
1479     }
1479
1480     }
1480
1481     }
1481
1482     }
1482
1483     }
1483
1484     }
1484
1485     }
1485
1486     }
1486
1487     }
1487
1488     }
1488
1489     }
1489
1490     }
1490
1491     }
1491
1492     }
1492
1493     }
1493
1494     }
1494
1495     }
1495
1496     }
1496
1497     }
1497
1498     }
1498
1499     }
1499
1500     }
1500
1501     }
1501
1502     }
1502
1503     }
1503
1504     }
1504
1505     }
1505
1506     }
1506
1507     }
1507
1508     }
1508
1509     }
1509
1510     }
1510
1511     }
1511
1512     }
1512
1513     }
1513
1514     }
1514
1515     }
1515
1516     }
1516
1517     }
1517
1518     }
1518
1519     }
1519
1520     }
1520
1521     }
1521
1522     }
1522
1523     }
1523
1524     }
1524
1525     }
1525
1526     }
1526
1527     }
1527
1528     }
1528
1529     }
1529
1530     }
1530
1531     }
1531
1532     }
1532
1533     }
1533
1534     }
1534
1535     }
1535
1536     }
1536
1537     }
1537
1538     }
1538
1539     }
1539
1540     }
1540
1541     }
1541
1542     }
1542
1543     }
1543
1544     }
1544
1545     }
1545
1546     }
1546
1547     }
1547
1548     }
1548
1549     }
1549
1550     }
1550
1551     }
1551
1552     }
1552
1553     }
1553
1554     }
1554
1555     }
1555
1556     }
1556
1557     }
1557
1558     }
1558
1559     }
1559
1560     }
1560
1561     }
1561
1562     }
1562
1563     }
1563
1564     }
1564
1565     }
1565
1566     }
1566
1567     }
1567
1568     }
1568
1569     }
1569
1570     }
1570
1571     }
1571
1572     }
1572
1573     }
1573
1574     }
1574
1575     }
1575
1576     }
1576
1577     }
1577
1578     }
1578
1579     }
1579
1580     }
1580
1581     }
1581
1582     }
1582
1583     }
1583
1584     }
1584
1585     }
1585
1586     }
1586
1587     }
1587
1588     }
1588
1589     }
1589
1590     }
1590
1591     }
1591
1592     }
1592
1593     }
1593
1594     }
1594
1595     }
1595
1596     }
1596
1597     }
1597
1598     }
1598
1599     }
1599
1600     }
1600
1601     }
1601
1602     }
1602
1603     }
1603
1604     }
1604
1605     }
1605
1606     }
1606
1607     }
1607
1608     }
1608
1609     }
1609
1610     }
1610
1611     }
1611
1612     }
1612
1613     }
1613
1614     }
1614
1615     }
1615
1616     }
1616
1617     }
1617
1618     }
1618
1619     }
1619
1620     }
1620
1621     }
1621
1622     }
1622
1623     }
1623
1624     }
1624
1625     }
1625
1626     }
1626
1627     }
1627
1628     }
1628
1629     }
1629
1630     }
1630
1631     }
1631
1632     }
1632
1633     }
1633
1634     }
1634
1635     }
1635
1636     }
1636
1637     }
1637
1638     }
1638
1639     }
1639
1640     }
1640
1641     }
1641
1642     }
1642
1643     }
1643
1644     }
1644
1645     }
1645
1646     }
1646
1647     }
1647
1648     }
1648
1649     }
1649
1650     }
1650
1651     }
1651
1652     }
1652
1653     }
1653
1654     }
1654
1655     }
1655
1656     }
1656
1657     }
1657
1658     }
1658
1659     }
1659
1660     }
1660
1661     }
1661
1662     }
1662
1663     }
1663
1664     }
1664
1665     }
1665
1666     }
1666
1667     }
1667
1668     }
1668
1669     }
1669
1670     }
1670
1671     }
1671
1672     }
1672
1673     }
1673
1674     }
1674
1675     }
1675
1676     }
1676
1677     }
1677
1678     }
1678
1679     }
1679
1680     }
1680
1681     }
1681
1682     }
1682
1683     }
1683
1684     }
1684
1685     }
1685
1686     }
1686
1687     }
1687
1688     }
1688
1689     }
1689
1690     }
1690
1691     }
1691
1692     }
1692
1693     }
1693
1694     }
1694
1695     }
1695
1696     }
1696
1697     }
1697
1698     }
1698
1699     }
1699
1700     }
1700
1701     }
1701
1702     }
1702
1703     }
1703
1704     }
1704
1705     }
1705
1706     }
1706
1707     }
1707
1708     }
1708
1709     }
1709
1710     }
1710
1711     }
1711
1712     }
1712
1713     }
1713
1714     }
1714
1715     }
1715
1716     }
1716
1717     }
1717
1718     }
1718
1719     }
1719
1720     }
1720
1721     }
1721
1722     }
1722
1723     }
1723
1724     }
1724
1725     }
1725
1726     }
1726
1727     }
1727
1728     }
1728
1729     }
1729
1730     }
1730
1731     }
1731
1732     }
1732
1733     }
1733
1734     }
1734
1735     }
1735
1736     }
1736
1737     }
1737
1738     }
1738
1739     }
1739
1740     }
1740
1741     }
1741
1742     }
1742
1743     }
1743
1744     }
1744
1745     }
1745
1746     }
1746
1747     }
1747
1748     }
1748
1749     }
1749
1750     }
1750
1751     }
1751
1752     }
1752
1753     }
1753
1754     }
1754
1755     }
1755
1756     }
1756
1757     }
1757
1758     }
1758
1759     }
1759
1760     }
1760
1761     }
1761
1762     }
1762
1763     }
1763
1764     }
1764
1765     }
1765
1766     }
1766
1767     }
1767
1768     }
1768
1769     }
1769
1770     }
1770
1771     }
1771
1772     }
1772
1773     }
1773
1774     }
1774
1775     }
1775
1776     }
1776
1777     }
1777
1778     }
1778
1779     }
1779
1780     }
1780
1781     }
1781
1782     }
1782
1783     }
1783
1784     }
1784
1785     }
1785
1786     }
1786
1787     }
1787
1788     }
1788
1789     }
1789
1790     }
1790
1791     }
1791
1792     }
1792
1793     }
1793
1794     }
1794
1795     }
1795
1796     }
1796
1797     }
1797
1798     }
1798
1799     }
1799
1800     }
1800
1801     }
1801
1802     }
1802
1803     }
1803
1804     }
1804
1805     }
1805
1806     }
1806
1807     }
1807
1808     }
1808
1809     }
1809
1810     }
1810
1811     }
1811
1812     }
1812
1813     }
1813
1814     }
1814
1815     }
1815
1816     }
1816
1817     }
1817
1818     }
1818
1819     }
1819
1820     }
1820
1821     }
1821
1822     }
1822
1823     }
1823
1824     }
1824
1825     }
1825
1826     }
1826
1827     }
1827
1828     }
1828
1829     }
1829
1830     }
1830
1831     }
1831
1832     }
1832
1833     }
1833
1834     }
1834
1835     }
1835
1836     }
1836
1837     }
1837
1838     }
1838
1839     }
1839
1840     }
1840
1841     }
1841
1842     }
1842
1843     }
1843
1844     }
1844
1845     }
1845
1846     }
1846
1847     }
1847
1848     }
1848
1849     }
1849
1850     }
1850
1851     }
1851
1852     }
1852
1853     }
1853
1854     }
1854
1855     }
1855
1856     }
1856
1857     }
1857
1858     }
1858
1859     }
1859
1860     }
1860
1861     }
1861
1862     }
1862
1863     }
1863
1864     }
1864
1865     }
1865
1866     }
1866
1867     }
1867
1868     }
1868
1869     }
1869
1870     }
1870
1871     }
1871
1872     }
1872
1873     }
1873
1874     }
1874
1875     }
1875
1876     }
1876
1877     }
1877
1878     }
1878
1879     }
1879
1880     }
1880
1881     }
1881
1882     }
1882
1883     }
1883
1884     }
1884
1885     }
1885
1886     }
1886
1887     }
1887
1888     }
1888
1889     }
1889
1890     }
1890
1891     }
1891
1892     }
1892
1893     }
1893
1894     }
1894
1895     }
1895
1896     }
1896
1897     }
1897
1898     }
1898
1899     }
1899
1900     }
1900
1901     }
1901
1902     }
1902
1903    
```

```

340 -             }
341 -         }
342 -
343 -         $this->_reconfigure('socks',
344 -             '3proxy.cfg', '/usr/local/etc/3proxy/3proxy.pid',
345 -             '/usr/local/etc/3proxy/3proxy.cfg', $gatewaysData['data']
346 -             ['socks'], 10, 20);
347 -
348 -             /*
349 -             sleep(10);
350 -
351 -             file_put_contents('/usr/local/etc/3proxy/3proxy.cfg',
352 -                 $gatewaysData['data']['socks']);
353 -
354 -             $socksProcess =
355 -             array_filter(explode("\n", exec('ps -fC 3proxy.cfg')));
356 -
357 -             if (!empty($socksProcess[0])) {
358 -                 $splitSocksProcess =
359 -                 array_values(array_filter(explode(' ',
360 -                     $socksProcess[0])));
361 -
362 -                 if (
363 -                     !empty($splitSocksProcess[1]) &&
364 -                     is_numeric($splitSocksProcess[1])
365 -                 ) {
366 -                     $shell =
367 -                     '#!/bin/bash';
368 -                     $shell .= "\n" .
369 -                     'kill ' . ($socksAlive ? '-SIGUSR1 ' : '') .
370 -                     trim($splitSocksProcess[1]);
371 -                     unlink('/scripts/socks.sh');
372 -                     file_put_contents('/scripts/socks.sh', $shell);
373 -                     shell_exec('chmod
374 -                     +x /scripts/socks.sh');
375 -                     shell_exec('/scripts/./socks.sh');
376 -                 }
377 -
378 -                 if (!$socksAlive) {
379 -                     unlink('/usr/local/etc/3proxy/3proxy.pid');
380 -                     sleep(10);
381 -                     shell_exec('service 3proxy start');
382 -                 }
383 -             }
384 -
385 -             // Add raw 20 second delay between
386 -             SOCKS and HTTP reconfiguration as necessary
387 -             sleep(20);*/

```

456

```

456 file_put_contents('/usr/local/etc/3proxy/3proxy.cfg',
457     $gatewaysData['data']['socks']);

```

457

```

458 + // Reconfigure main SOCKS instance
459 +
460 + $this->_reconfigure(
461 +     'socks',
462 +     '3proxy',
463 +     null,
464 +     null,
465 +     '3proxy.cfg',
466 +     '/usr/local/etc/3proxy/3proxy.pid',
467 +     10,
468 +     20
469 + );

```

```

373     }
374
375     /*
376     $mainResponse = shell_exec('squid3 -k
reconfigure');
377
378     if (strpos(strtolower($mainResponse), '
error:') !== false) {
379         exec('ps -fc squid3 2>&1',
$processes);
380
381         if (!empty($processes[0])) {
382             unset($processes[0]);
383
384             foreach ($processes as
$process) {
385                 $processColumns =
array_values(array_filter(explode(' ', $process)));
386
387                 if
(!empty($processColumns[1])) {
388                     $killProcesses[] = $processColumns[1];
389                 }
390             }
391
392             if
(!empty($killProcesses)) {
393                 $shell =
'#!/bin/bash';
394
395                 foreach
($killProcesses as $killProcess) {
396                     $shell .=
"\n" . 'kill ' . trim($killProcess);
397                 }
398
399                 unlink('/scripts/http.sh');
400
401                 file_put_contents('/scripts/http.sh', $shell);
402                 shell_exec('chmod
+x /scripts/http.sh');
403                 shell_exec('/scripts/./http.sh');
404             }

```

```

469     }
470
471     // Reconfigure main HTTP instance
472     $this->_reconfigure(
473         'http',
474         'squid3',
475         'squid3 -k reconfigure',
476         ' error:',
477         'squid3',
478         '/var/run/squid3.pid',
479         0,
480         70
481     );
482
483     $redundantProcesses = $this->
processes['http']; // TODO: chunk and reconfigure HTTP
and SOCKS processes simultaneously
484     unset($redundantProcesses[0]);
485     $redundantProcessChunks =
array_chunk($redundantProcesses, 5, true);
486
487     foreach ($redundantProcessChunks as
$redundantProcessChunk) {
488         // Define active redundant process
numbers
489         $activeRedundantProcesses =
array_keys($redundantProcesses);
490
491         // Define redundant process number
range for reconfiguration
492         reset($redundantProcessChunk);
493         $redundantProcessStart =
key($redundantProcessChunk);
494         end($redundantProcessChunk);
495         $redundantProcessEnd =
key($redundantProcessChunk);
496         $redundantProcessRange =
range($redundantProcessStart, $redundantProcessEnd);
497
498         foreach ($redundantProcessRange as
$redundantProcess) {
499             unset($activeRedundantProcesses[$redundantProcess - 1]);

```


<pre> 404 } 405 406 - unlink('/var/run/squid3.pid'); 407 - 408 sleep(3); 409 shell_exec('squid3 start'); 410 } 411 412 sleep(70); 413 unlink('/scripts/iptables/iptables- redundant'); 414 415 shell_exec('php /scripts/firewall_redundant.php ports6+ports7+ports8+ports9+ports10'); 416 417 foreach (range(1, 5) as \$instance) { 418 \$instance = \$instance == 1 ? '' : \$instance; 419 exec('squid3-redundant' . \$instance . ' -k reconfigure -f /etc/squid3/squid- redundant' . \$instance . '.conf 2>&1', \$redundantResponse); 420 \$redundantResponse = array_shift(\$redundantResponse); 421 422 if (strpos(strtolower(\$redundantResponse), ' error:') !== false) { 423 exec('ps -fc squid3- redundant 2>&1', \$processes); 424 425 if (!empty(\$processes[0])) { 426 unset(\$processes[0]); 427 428 foreach (\$processes as \$process) { 429 \$processColumns = array_values(array_filter(explode(' ', \$process))); 430 431 if (!empty(\$processColumns[1]) && in_array('/etc/squid3/squid-redundant' . \$instance . '.conf', \$processColumns) 432) { 433 \$killProcesses[] = \$processColumns[1]; 434 } 435 } 436 437 if (!empty(\$killProcesses)) { 438 \$shell = '#!/bin/bash'; 439 440 foreach (\$killProcesses as \$killProcess) { 441 \$shell .= "\n" . 'kill ' . trim(\$killProcess); </pre>	<pre> 500 } 501 502 + // Get list of active redundant ports for firewall configuration 503 + \$activeRedundantPorts = array(); </pre>
	<pre> 504 505 + foreach (\$redundantProcesses as \$key => \$redundantProcess) { 506 + if (in_array(\$key, \$activeRedundantProcesses)) { </pre>

<pre> 442 - } 443 - 444 - 445 - unlink('/scripts/http.sh'); 446 - 447 - file_put_contents('/scripts/http.sh', \$shell); 448 - 449 - shell_exec('chmod +x /scripts/http.sh'); 450 - 451 - shell_exec('/scripts/./http.sh'); 452 - } 453 - } 454 - 455 - shell_exec('rm /var/run/squid-redundant' . \$instance . '.pid'); 456 - sleep(3); 457 - shell_exec('squid3- redundant' . \$instance . ' start -f /etc/squid3/squid- redundant' . \$instance . '.conf'); 458 - } 459 - } 460 - 461 - sleep(70); 462 - shell_exec('php /scripts/firewall_redundant.php ports1+ports2+ports3+ports4+ports5'); 463 - 464 - foreach (range(6, 10) as \$instance) { 465 - exec('squid3-redundant' . \$instance . ' -k reconfigure -f /etc/squid3/squid- redundant' . \$instance . '.conf 2>&1', \$redundantResponse); 466 - \$redundantResponse = array_shift(\$redundantResponse); 467 - 468 - if (strpos(strtolower(\$redundantResponse), ' error:') !== false) { 469 - exec('ps -fC squid3- redundant 2>&1', \$processes); 470 - 471 - if (!empty(\$processes[0])) { 472 - 473 - unset(\$processes[0]); 474 - 475 - foreach (\$processes as \$process) { 476 - 477 - \$processColumns = array_values(array_filter(explode(' ', \$process))); 478 - 479 - if (!empty(\$processColumns[1]) && in_array('/etc/squid3/squid-redundant' . \$instance . '.conf', \$processColumns) 480 -) { 481 - 482 - \$killProcesses[] = \$processColumns[1]; </pre>	<pre> 507 + \$activeRedundantPorts = array_merge(\$activeRedundantPorts, \$redundantProcesses[\$key]); 508 + 509 + 510 + 511 + \$overridePorts = array(512 + 'http' => 513 + array_merge(array(514 + '80', '8888', 515 + '55555' 516 +), \$activeRedundantPorts), 517 + 'socks' => array(518 + '1090' 519 +) 520 +); 521 + </pre>
--	---

```

478 -                                     }
479 -                                     }
480 -
481 -                                     if
482 -                                     (!empty($killProcesses)) {
483 -                                     $shell =
484 -                                     '#!/bin/bash';
485 -                                     foreach
486 -                                     ($killProcesses as $killProcess) {
487 -                                     $shell .= "\n" . 'kill ' . trim($killProcess);
488 -                                     }
489 -                                     }
490 -                                     }
491 -                                     }
492 -                                     }
493 -                                     }
494 -                                     }
495 -                                     shell_exec('rm
496 -                                     /var/run/squid-redundant' . $instance . '.pid');
497 -                                     sleep(3);
498 -                                     shell_exec('squid3-
499 -                                     redundant' . $instance . ' start -f /etc/squid3/squid-
500 -                                     redundant' . $instance . '.conf');
501 -                                     }
502 -                                     }
503 -                                     }
504 -                                     }
505 -                                     }
506 -                                     }
507 -                                     }
508 -                                     }
509 -                                     }
510 -                                     }
511 -                                     }
512 -                                     }
513 -                                     }
514 -                                     }
515 -                                     }
516 -                                     }
517 -                                     }
518 -                                     }
519 -                                     }
520 -                                     }
521 -                                     }
522 -                                     }
523 -                                     }
524 -                                     }
525 -                                     }
526 -                                     }
527 -                                     }
528 -                                     }
529 -                                     }
530 -                                     }
531 -                                     }
532 -                                     }
533 -                                     }
534 -                                     }
535 -                                     }
536 -                                     }
537 -                                     }
538 -                                     }
539 -                                     }
540 -                                     }
541 -                                     }
542 -                                     }
543 -                                     }
544 -                                     }
545 -                                     }
546 -                                     }
547 -                                     }
548 -                                     }
549 -                                     }
550 -                                     }
551 -                                     }
552 -                                     }
553 -                                     }
554 -                                     }
555 -                                     }
556 -                                     }
557 -                                     }
558 -                                     }
559 -                                     }
560 -                                     }
561 -                                     }
562 -                                     }
563 -                                     }
564 -                                     }
565 -                                     }
566 -                                     }
567 -                                     }
568 -                                     }
569 -                                     }
570 -                                     }
571 -                                     }
572 -                                     }
573 -                                     }
574 -                                     }
575 -                                     }
576 -                                     }
577 -                                     }
578 -                                     }
579 -                                     }
580 -                                     }
581 -                                     }
582 -                                     }
583 -                                     }
584 -                                     }
585 -                                     }
586 -                                     }
587 -                                     }
588 -                                     }
589 -                                     }
590 -                                     }
591 -                                     }
592 -                                     }
593 -                                     }
594 -                                     }
595 -                                     }
596 -                                     }
597 -                                     }
598 -                                     }
599 -                                     }
600 -                                     }
601 -                                     }
602 -                                     }
603 -                                     }
604 -                                     }
605 -                                     }
606 -                                     }
607 -                                     }
608 -                                     }
609 -                                     }
610 -                                     }
611 -                                     }
612 -                                     }
613 -                                     }
614 -                                     }
615 -                                     }
616 -                                     }
617 -                                     }
618 -                                     }
619 -                                     }
620 -                                     }
621 -                                     }
622 -                                     }
623 -                                     }
624 -                                     }
625 -                                     }
626 -                                     }
627 -                                     }
628 -                                     }
629 -                                     }
630 -                                     }
631 -                                     }
632 -                                     }
633 -                                     }
634 -                                     }
635 -                                     }
636 -                                     }
637 -                                     }
638 -                                     }
639 -                                     }
640 -                                     }
641 -                                     }
642 -                                     }
643 -                                     }
644 -                                     }
645 -                                     }
646 -                                     }
647 -                                     }
648 -                                     }
649 -                                     }
650 -                                     }
651 -                                     }
652 -                                     }
653 -                                     }
654 -                                     }
655 -                                     }
656 -                                     }
657 -                                     }
658 -                                     }
659 -                                     }
660 -                                     }
661 -                                     }
662 -                                     }
663 -                                     }
664 -                                     }
665 -                                     }
666 -                                     }
667 -                                     }
668 -                                     }
669 -                                     }
670 -                                     }
671 -                                     }
672 -                                     }
673 -                                     }
674 -                                     }
675 -                                     }
676 -                                     }
677 -                                     }
678 -                                     }
679 -                                     }
680 -                                     }
681 -                                     }
682 -                                     }
683 -                                     }
684 -                                     }
685 -                                     }
686 -                                     }
687 -                                     }
688 -                                     }
689 -                                     }
690 -                                     }
691 -                                     }
692 -                                     }
693 -                                     }
694 -                                     }
695 -                                     }
696 -                                     }
697 -                                     }
698 -                                     }
699 -                                     }
700 -                                     }
701 -                                     }
702 -                                     }
703 -                                     }
704 -                                     }
705 -                                     }
706 -                                     }
707 -                                     }
708 -                                     }
709 -                                     }
710 -                                     }
711 -                                     }
712 -                                     }
713 -                                     }
714 -                                     }
715 -                                     }
716 -                                     }
717 -                                     }
718 -                                     }
719 -                                     }
720 -                                     }
721 -                                     }
722 -                                     }
723 -                                     }
724 -                                     }
725 -                                     }
726 -                                     }
727 -                                     }
728 -                                     }
729 -                                     }
730 -                                     }
731 -                                     }
732 -                                     }
733 -                                     }
734 -                                     }
735 -                                     }
736 -                                     }
737 -                                     }
738 -                                     }
739 -                                     }
740 -                                     }
741 -                                     }
742 -                                     }
743 -                                     }
744 -                                     }
745 -                                     }
746 -                                     }
747 -                                     }
748 -                                     }
749 -                                     }
750 -                                     }
751 -                                     }
752 -                                     }
753 -                                     }
754 -                                     }
755 -                                     }
756 -                                     }
757 -                                     }
758 -                                     }
759 -                                     }
760 -                                     }
761 -                                     }
762 -                                     }
763 -                                     }
764 -                                     }
765 -                                     }
766 -                                     }
767 -                                     }
768 -                                     }
769 -                                     }
770 -                                     }
771 -                                     }
772 -                                     }
773 -                                     }
774 -                                     }
775 -                                     }
776 -                                     }
777 -                                     }
778 -                                     }
779 -                                     }
780 -                                     }
781 -                                     }
782 -                                     }
783 -                                     }
784 -                                     }
785 -                                     }
786 -                                     }
787 -                                     }
788 -                                     }
789 -                                     }
790 -                                     }
791 -                                     }
792 -                                     }
793 -                                     }
794 -                                     }
795 -                                     }
796 -                                     }
797 -                                     }
798 -                                     }
799 -                                     }
800 -                                     }
801 -                                     }
802 -                                     }
803 -                                     }
804 -                                     }
805 -                                     }
806 -                                     }
807 -                                     }
808 -                                     }
809 -                                     }
810 -                                     }
811 -                                     }
812 -                                     }
813 -                                     }
814 -                                     }
815 -                                     }
816 -                                     }
817 -                                     }
818 -                                     }
819 -                                     }
820 -                                     }
821 -                                     }
822 -                                     }
823 -                                     }
824 -                                     }
825 -                                     }
826 -                                     }
827 -                                     }
828 -                                     }
829 -                                     }
830 -                                     }
831 -                                     }
832 -                                     }
833 -                                     }
834 -                                     }
835 -                                     }
836 -                                     }
837 -                                     }
838 -                                     }
839 -                                     }
840 -                                     }
841 -                                     }
842 -                                     }
843 -                                     }
844 -                                     }
845 -                                     }
846 -                                     }
847 -                                     }
848 -                                     }
849 -                                     }
850 -                                     }
851 -                                     }
852 -                                     }
853 -                                     }
854 -                                     }
855 -                                     }
856 -                                     }
857 -                                     }
858 -                                     }
859 -                                     }
860 -                                     }
861 -                                     }
862 -                                     }
863 -                                     }
864 -                                     }
865 -                                     }
866 -                                     }
867 -                                     }
868 -                                     }
869 -                                     }
870 -                                     }
871 -                                     }
872 -                                     }
873 -                                     }
874 -                                     }
875 -                                     }
876 -                                     }
877 -                                     }
878 -                                     }
879 -                                     }
880 -                                     }
881 -                                     }
882 -                                     }
883 -                                     }
884 -                                     }
885 -                                     }
886 -                                     }
887 -                                     }
888 -                                     }
889 -                                     }
890 -                                     }
891 -                                     }
892 -                                     }
893 -                                     }
894 -                                     }
895 -                                     }
896 -                                     }
897 -                                     }
898 -                                     }
899 -                                     }
900 -                                     }
901 -                                     }
902 -                                     }
903 -                                     }
904 -                                     }
905 -                                     }
906 -                                     }
907 -                                     }
908 -                                     }
909 -                                     }
910 -                                     }
911 -                                     }
912 -                                     }
913 -                                     }
914 -                                     }
915 -                                     }
916 -                                     }
917 -                                     }
918 -                                     }
919 -                                     }
920 -                                     }
921 -                                     }
922 -                                     }
923 -                                     }
924 -                                     }
925 -                                     }
926 -                                     }
927 -                                     }
928 -                                     }
929 -                                     }
930 -                                     }
931 -                                     }
932 -                                     }
933 -                                     }
934 -                                     }
935 -                                     }
936 -                                     }
937 -                                     }
938 -                                     }
939 -                                     }
940 -                                     }
941 -                                     }
942 -                                     }
943 -                                     }
944 -                                     }
945 -                                     }
946 -                                     }
947 -                                     }
948 -                                     }
949 -                                     }
950 -                                     }
951 -                                     }
952 -                                     }
953 -                                     }
954 -                                     }
955 -                                     }
956 -                                     }
957 -                                     }
958 -                                     }
959 -                                     }
960 -                                     }
961 -                                     }
962 -                                     }
963 -                                     }
964 -                                     }
965 -                                     }
966 -                                     }
967 -                                     }
968 -                                     }
969 -                                     }
970 -                                     }
971 -                                     }
972 -                                     }
973 -                                     }
974 -                                     }
975 -                                     }
976 -                                     }
977 -                                     }
978 -                                     }
979 -                                     }
980 -                                     }
981 -                                     }
982 -                                     }
983 -                                     }
984 -                                     }
985 -                                     }
986 -                                     }
987 -                                     }
988 -                                     }
989 -                                     }
990 -                                     }
991 -                                     }
992 -                                     }
993 -                                     }
994 -                                     }
995 -                                     }
996 -                                     }
997 -                                     }
998 -                                     }
999 -                                     }
1000 -                                     }

```

```

506 -             shell_exec('php
/scripts/firewall.php');
507 -             exec('curl --socks5-hostname ' .
$proxies[0] . ':1090' . ' http://socks/ -v --connect-
timeout 4 --max-time 4 2>&1', $socksResponseRedundant);
508 -             $socksResponseRedundant =
end($socksResponseRedundant);
509 -             $socksAliveRedundant =
(strpos(strtolower($socksResponseRedundant), 'empty reply
') !== false);
510 -
file_put_contents('/usr/local/etc/3proxy/3proxy-
redundant.cfg', $gatewaysData['data']['socks-redundant']);
511 -             $socksProcessRedundant =
array_filter(explode("\n", exec('ps -fC 3proxy-
redundant.cfg')));
512 -
513 -             if
(!empty($socksProcessRedundant[0])) {
514 -
$splitSocksProcessRedundant =
array_values(array_filter(explode(' ',
$socksProcessRedundant[0])));
515 -
516 -                 if (
517 -
!empty($splitSocksProcessRedundant[1]) &&
518 -
is_numeric($splitSocksProcessRedundant[1])
519 -
) {
520 -
$shell =
'#!/bin/bash';
521 -
$shell .= "\n" .
'kill ' . ($socksAliveRedundant ? '-SIGUSR1 ' : '') .
trim($splitSocksProcessRedundant[1]);
522 -
unlink('/scripts/socks.sh');
523 -
file_put_contents('/scripts/socks.sh', $shell);
524 -
shell_exec('chmod
+x /scripts/socks.sh');
525 -
shell_exec('/scripts./socks.sh');
526 -
}
527 -
528 -             if (!$socksAliveRedundant)
529 -
{
530 -
sleep(10);
531 -
shell_exec('service 3proxy-redundant start');
532 -
}
533 -
}

534 -
}

535 -
536 -             shell_exec('php /scripts/firewall.php

```

```

552 +
553 + // Reconfigure redundant SOCKS
instance
554 + $this->_reconfigure(
'socks',
's3proxy-redundant',
555 +
null,
556 +
null,
557 +
'3proxy-redundant.cfg',
558 +
'/usr/local/etc/3proxy/3proxy-redundant.pid',
559 +
15,
560 +
0
561 +
);
562 +
563 +
564 +
565 +
566 + $this->_applyFirewallRules($firewallRules,

```

<pre> 537 unlink('/scripts/pid/reconfigure.pid'); 538 - die(); 539 - */ 540 } 541 542 /** 586 } 587 588 } 589 - 590 - // Debug code, remove later 591 - \$api = 'https://ghostproxies.com/gateways/api.json'; 592 - \$processes = array(593 - 'http' => array(594 - array(595 - '80', 596 - '8888', 597 - '55555'), 598 - array(599 - '4444', 600 - '7777' 601 -), 602 - array(603 - '55540', 604 - '55541' 605 -), 606 - array(607 - '55546', 608 - '55547' 609 -), 610 - array(611 - '55520', 612 - '55521' 613 -), 614 - array(615 - '55524', 616 - '55525' 617 -), 618 - array(619 - '55526', 620 - '55527' 621 -), 622 - array(623 - '55528', 624 - '55529' 625 -), 626 - array(627 - '55530', 628 - '55531' 629 -), 630 - array(631 - '55532', 632 - '55533' 633 -), 634 - array(635 - '55534', 636 - '55535' </pre>	<pre> 'elastic'); 567 + 568 + // Remove reconfiguration process ID 569 + unlink('/scripts/pid/reconfigure.pid'); 570 + 571 + // Memory cleanup 572 + gc_collect_cycles(); 573 + 574 + return true; 575 } 576 577 /** 621 } 622 623 } </pre>
--	---

```

637 -         )
638 -     ),
639 -     'socks' => array(
640 -         array(
641 -             '1085'
642 -         ),
643 -         array(
644 -             '1090'
645 -         )
646 -     )
647 - );
648 - $sshPorts = array(
649 -     '22',
650 -     '5000'
651 - );
652 -
653 - $dynamicProxyReconfiguration = new
DynamicProxyReconfiguration($api, $processes, $sshPorts);
654 - $status = $dynamicProxyReconfiguration-
>start('apply_reconfiguration');
655 -
656 - print_r($status);
657 - die();
658 - ?>
659 -
660 - TODO:
661 - Add support for additional HTTP and SOCKS proxy servers
662 - Apply proxy redundancy configuration from API to all
redundant processes
663 - Start additional redundancy processes when CPU exceeds
x%, provide option to terminate redundant processes if
average CPU usage is below x%

```

624

?>

0 comments on commit `d09e8c2`