

documatação pyFirewall

JUAN CARLOS BINDEZ

Versão v1.1.1

Quarta, 24 de Agosto de 2022

Versão Atual pyFirewall v1.1.1

"This project is licensed under the MIT License."

"Este projeto está licenciado nos termos da licença MIT."

Detalhes de lançamento

v1.1.1:

adicionado regras de bloqueios e liberação de IPs específicos
correção de bugs

v1.0.4:

alterações na interface do usuário

v1.0.3:

correção de bugs
reestruturação de código
adicionado recursos de melhorias de navegação pelo menu.

v1.0.2:

adicionado recurso para salvar as alterações do firewall
deletar regras

v1.0.1:

reestruturação de código
correção de falhas

v1.0.0:

versão inicial

Objetivo do software:

facilitar a configuração de firewall (iptables)
diminuir a quantidade de comandos digitados
facilitar a visualização das regras e diminuir o tempo de configuração de firewall

o que é o pyFirewall:

O pyFirewall é um software escrito em Python na versão 3.10.4, que visa manipular os comandos do iptables (<https://g.co/kgs/9ZJDYt>), este programa pode te ajudar a entender as regras de firewall e facilitar as configurações.

Como usar?

Faça um git clone:

```
git clone https://github.com/JuanBindez/pyFirewall-v1.1.1
```

Acesse a pasta:

```
cd pyFirewall-v1.1.1/
```

Agora é só rodar o software:

```
python3 pyfirewall.py
```

Índice dos namespaces

Lista de namespaces

Lista dos namespaces com uma breve descrição:

banner	5
colors	6
ipv4	6
ipv4.logica_ipv4	6
ipv6	Erro: Origem da referência não encontrada
pyfirewall	7

Índice dos componentes

Lista de componentes

Lista de classes, estruturas, uniões e interfaces com uma breve descrição:

colors.Color	14
ipv4.logica_ipv4.DeleteRegra	15
ipv4.logica_ipv4.IpRegras	16
ipv4.logica_ipv4.LogicasMenu1	17
ipv4.logica_ipv4.RegrasList	19
ipv4.logica_ipv4.SaveTable	20

Índice dos ficheiros

Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

pyFirewall-v1.1.1/banner.py	21
pyFirewall-v1.1.1/colors.py	22
pyFirewall-v1.1.1/pyfirewall.py	23
pyFirewall-v1.1.1/ipv4/__init__.py	22
pyFirewall-v1.1.1/ipv4/logica_ipv4.py	22
pyFirewall-v1.1.1/ipv6/__init__.py	22

Referência ao namespace colors

Componentes

class **Color**

Descrição detalhada

classe para colorir a interface do usuario.

Referência ao namespace ipv4

Namespaces

logica_ipv4

Descrição detalhada

Copyright (c) 2022 Juan Carlos Bindez
"This project is licensed under the MIT License."

Referência ao namespace ipv4.logica_ipv4

Componentes

class **LogicasMenu1**
class **RegrasList**
class **DeleteRegra**
class **SaveTable**
class **IpRegras**

Descrição detalhada

Classes usadas para armazenar comandos do iptables,
e metodos que executam os comandos.

Referência ao namespace ipv6

Descrição detalhada
ipv6 ainda não implementado.

Referência ao namespace pyfirewall

Funções

def menu_main_ipv4 ()

Todas as funções estão dentro desta.

Variáveis

ver_regras = LogicasMenu1("sudo iptables -L --line-numbers")

delete = LogicasMenu1("sudo iptables -D INPUT")

Descrição detalhada

Todas as funções desse script buscam comandos embutidos nas classes do arquivo logica_ipv4.py e os executam, conforme solicitado na interface do usuário.

Documentação das funções

def pyfirewall.menu_main_ipv4 ()

INICIO DO BLOCO DE MENU IPV4 #####.

```
31     def menu_main_ipv4():
32
33
34         def Ver_regras_firewall():
35             # ve regras existentes no firewall
36             os.system("clear")
37             ver_regras.start_command()
38             menu_main_ipv4()
39
40
41         def deletar_regras_firewall():
42             os.system("clear")
43             ver_regras.start_command()
44             header_banner()
45             # deleta id de regra no firewall
46             print(Color.AMARELO +
47
48                 '''
49
50                                     Deletar de qual tabela?
51
52                                     *[0]Voltar
53                                     *[1]INPUT
54                                     *[2]FORWARD
55                                     *[3]OUTPUT
56
57                 '''
58             + Color.RESET)
59
60             choice_delete = str(input(">>"))
61
62             if choice_delete == "0":
63                 os.system("clear")
64                 menu_main_ipv4()
65
66             elif choice_delete == "1":
67                 DeleteRegra.delete_INPUT.delete_id()
68                 os.system("clear")
```

```

68         ver_regras.start_command()
69         menu_main_ipv4()
70
71     elif choice_delete == "2":
72         DeleteRegra.delete_FORWARD.delete_id()
73         os.system("clear")
74         ver_regras.start_command()
75         menu_main_ipv4()
76
77     elif choice_delete == "3":
78         DeleteRegra.delete_OUTPUT.delete_id()
79         os.system("clear")
80         ver_regras.start_command()
81         menu_main_ipv4()
82
83     else:
84         os.system("clear")
85         print("ops, digite apenas os numeros listados!")
86         menu_main_ipv4()
87
88     os.system("clear")
89     ver_regras.start_command()
90     delete.delete_id()
91     menu_main_ipv4()
92
93
94
95 def regras_de_ports_firewall():
96
97
98     def regra_port_INPUT():
99         header_regra_port()
100         choice_regra = str(input(">>"))
101
102         if choice_regra == "0":
103             os.system("clear")
104             menu_main_ipv4()
105
106         elif choice_regra == "1":
107             RegrasList.ports_tab_input_accept.port_change()
108             os.system("clear")
109             ver_regras.start_command()
110             menu_main_ipv4()
111
112         elif choice_regra == "2":
113             RegrasList.ports_tab_input_drop.port_change()
114             os.system("clear")
115             ver_regras.start_command()
116             menu_main_ipv4()
117
118         else:
119             os.system("clear")
120             print("ops, digite apenas os numeros listados!")
121             menu_main_ipv4()
122
123
124     def header_regra_port():
125         os.system("clear")
126         header_banner()
127         print(Color.AMARELO +
128               '''
129                               *[0]Voltar
130                               *[1]ACCEPT
131                               *[2]DROP
132
133               ''')
134         + Color.RESET)
135
136
137
138     def regra_port_FORWARD():
139         header_regra_port()
140         choice_regra = str(input(">>"))
141
142         if choice_regra == "1":
143             RegrasList.ports_tab_forward_accept.port_change()
144             os.system("clear")

```



```

145         ver_regras.start_command()
146         menu_main_ipv4()
147
148     elif choice_regra == "2":
149         RegrasList.ports_tab_forward_drop.port_change()
150         os.system("clear")
151         ver_regras.start_command()
152         menu_main_ipv4()
153
154     else:
155         os.system("clear")
156         print("ops, digite apenas os numeros listados!")
157         menu_main_ipv4()
158
159
160 def regra_port_OUTPUT():
161     header_regra_port()
162     choice_regra = str(input(">>"))
163
164     if choice_regra == "1":
165         RegrasList.ports_tab_output_accept.port_change()
166         os.system("clear")
167         ver_regras.start_command()
168         menu_main_ipv4()
169
170     elif choice_regra == "2":
171         RegrasList.ports_tab_output_drop.port_change()
172         os.system("clear")
173         ver_regras.start_command()
174         menu_main_ipv4()
175
176     else:
177         os.system("clear")
178         print("ops, digite apenas os numeros listados!")
179         menu_main_ipv4()
180
181
182
183 os.system("clear")
184 ver_regras.start_command()
185 header_banner()
186 print(Color.AMARELO +
187       '''
188
189                                     Escolha a Tabela
190
191                                     *[0]Voltar
192                                     *[1]INPUT
193                                     *[2]FORWARD
194                                     *[3]OUTPUT
195
196       ''')
197
198 choice_tab = str(input(">>>"))
199
200 if choice_tab == "0":
201     os.system("clear")
202     menu_main_ipv4()
203
204 elif choice_tab == "1":
205     regra_port_INPUT()
206
207 elif choice_tab == "2":
208     regra_port_FORWARD()
209
210 elif choice_tab == "3":
211     regra_port_OUTPUT()
212
213
214
215
216
217 def salva_regras_firewall():
218     os.system("sudo service netfilter-persistent save")
219     time.sleep(2)
220     os.system("sudo systemctl restart netfilter-
221 persistent.service")
222     time.sleep(2)

```

```

222         os.system("sudo systemctl status netfilter-persistent.service")
223         os.system("clear")
224         menu_main_ipv4()
225
226
227     def netfilter_install():
228         os.system("sudo apt-get install netfilter-persistent.service")
229         os.system("sudo apt-get install iptables-persistent")
230         time.sleep(2)
231         os.system("clear")
232         menu_main_ipv4()
233
234
235     def exclui_tab_firewall():
236         os.system("clear")
237         ver_regras.start_command()
238         header_banner()
239
240         print(Color.AMARELO +
241
242             '''
243                                     Escolha a Tabela a ser
Excluída
244
245                                     *[0]Voltar
246                                     *[1]INPUT
247                                     *[2]FORWARD
248                                     *[3]OUTPUT
249                                     *[4]Todas as tabelas
250
251             '''
252         + Color.RESET)
253
254         escolha = str(input(">>"))
255
256         if escolha == "0":
257             os.system("clear")
258             menu_main_ipv4()
259
260         elif escolha == "1":
261             os.system("sudo iptables -F INPUT")
262             os.system("clear")
263             ver_regras.start_command()
264             menu_main_ipv4()
265
266         elif escolha == "2":
267             os.system("sudo iptables -F FORWARD")
268             os.system("clear")
269             ver_regras.start_command()
270             menu_main_ipv4()
271
272         elif escolha == "3":
273             os.system("sudo iptables -F OUTPUT")
274             os.system("clear")
275             ver_regras.start_command()
276             menu_main_ipv4()
277
278         elif escolha == "4":
279             os.system("sudo iptables -F")
280             os.system("clear")
281             ver_regras.start_command()
282             menu_main_ipv4()
283
284         else:
285             os.system("clear")
286             print("ops, digite apenas os numeros listados!")
287             menu_main_ipv4()
288
289
290
291
292     def ip_regras():
293
294         def header_escolha7():
295             os.system("clear")
296             header_banner()
297

```

```

298         print(Color.AMARELO +
299             '''
300                                     *[0]Voltar
301                                     *[1]ACCEPT
302                                     *[2]DROP
303
304             '''
305         + Color.RESET)
306
307
308
309     def ip_regra_INPUT_ACCEPT():
310         IpRegras.ip_ACCEPT_tab_INPUT.ip_func_regra()
311         os.system("clear")
312         ver_regras.start_command()
313         menu_main_ipv4()
314
315
316     def ip_regra_FORWARD_ACCEPT():
317         IpRegras.ip_ACCEPT_tab_FORWARD.ip_func_regra()
318         os.system("clear")
319         ver_regras.start_command()
320         menu_main_ipv4()
321
322
323
324     def ip_regra_OUTPUT_ACCEPT():
325         IpRegras.ip_ACCEPT_tab_OUTPUT.ip_func_regra()
326         os.system("clear")
327         ver_regras.start_command()
328         menu_main_ipv4()
329
330
331
332     def ip_regra_INPUT_DROP():
333         IpRegras.ip_DROP_tab_INPUT.ip_func_regra()
334         os.system("clear")
335         ver_regras.start_command()
336         menu_main_ipv4()
337
338
339     def ip_regra_FORWARD_DROP():
340         IpRegras.ip_DROP_tab_FORWARD.ip_func_regra()
341         os.system("clear")
342         ver_regras.start_command()
343         menu_main_ipv4()
344
345
346     def ip_regra_OUTPUT_DROP():
347         IpRegras.ip_DROP_tab_OUTPUT.ip_func_regra()
348         os.system("clear")
349         ver_regras.start_command()
350         menu_main_ipv4()
351
352
353     os.system("clear")
354     header_banner()
355     print(Color.AMARELO +
356         '''
357
358                                     Escolha a Tabela
359
360                                     *[0]Voltar
361                                     *[1]INPUT
362                                     *[2]FORWARD
363                                     *[3]OUTPUT
364
365         '''
366     + Color.RESET)
367
368     escolha7 = str(input(">>"))
369
370     if escolha7 == "0":
371         os.system("clear")
372         menu_main_ipv4()
373
374     elif escolha7 == "1":
375         header_escolha7()

```

```

375         escolha = str(input(">>"))
376
377         if escolha == "0":
378             os.system("clear")
379             menu_main_ipv4()
380
381         elif escolha == "1":
382             ip_regra_INPUT_ACCEPT()
383
384         elif escolha == "2":
385             ip_regra_INPUT_DROP()
386
387
388     elif escolha7 == "2":
389         header_escolha7()
390         escolha = str(input(">>"))
391
392         if escolha == "0":
393             os.system("clear")
394             menu_main_ipv4()
395
396         elif escolha == "1":
397             ip_regra_FORWARD_ACCEPT()
398
399         elif escolha == "2":
400             ip_regra_FORWARD_DROP()
401
402     elif escolha7 == "3":
403         header_escolha7()
404         escolha = str(input(">>"))
405
406         if escolha == "0":
407             os.system("clear")
408             menu_main_ipv4()
409
410         elif escolha == "1":
411             ip_regra_OUTPUT_ACCEPT()
412
413         elif escolha == "2":
414             ip_regra_OUTPUT_DROP()
415
416     else:
417         os.system("clear")
418         print("Ops, Digite apenas os numeros listados!")
419         ip_regras()
420
421
422
423
425
426
427
428
429     header_banner()
430     print(Color.AMARELO +
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
'''

```

```
452         regras_de_ports_firewall()
453
454     elif choice == "4":
455         salva_regras_firewall()
456
457     elif choice == "5":
458         netfilter_install()
459
460     elif choice == "6":
461         exclui_tab_firewall()
462
463     elif choice == "7":
464         ip_regras()
465
466     else:
467         os.system("clear")
468         print("Digite Apenas os Números Listados!")
469         menu_main_ipv4()
470
471
472
```

Documentação das variáveis

pyfirewall.delete = LogicasMenu1("sudo iptables -D INPUT")

pyfirewall.ver_regras = LogicasMenu1("sudo iptables -L --line-numbers")

Documentação da classe

Referência à classe colors.Color

Atributos Públicos Estáticos

```
string VERDE = '\033[92m'  
string VERDE_CLARO = '\033[1;92m'  
string VERMELHO = '\033[91m'  
string AMARELO = '\033[93m'  
string AZUL = '\033[1;34m'  
string MAGENTA = '\033[1;35m'  
string NEGRITO = '\033[1m'  
string CYANO = '\033[1;36m'  
string CYANO_CLARO = '\033[1;96m'  
string CINZA_CLARO = '\033[1;37m'  
string CINZA_ESCURO = '\033[1;90m'  
string PRETO = '\033[1;30m'  
string BRANCO = '\033[1;97m'  
string INVERTE = '\033[7m'  
string RESET = '\033[0m'
```

Documentação dos dados membro

```
string colors.Color.AMARELO = '\033[93m'[static]

string colors.Color.AZUL = '\033[1;34m'[static]

string colors.Color.BRANCO = '\033[1;97m'[static]

string colors.Color.CINZA_CLARO = '\033[1;37m'[static]

string colors.Color.CINZA_ESCURO = '\033[1;90m'[static]

string colors.Color.CYANO = '\033[1;36m'[static]

string colors.Color.CYANO_CLARO = '\033[1;96m'[static]

string colors.Color.INVERTE = '\033[7m'[static]

string colors.Color.MAGENTA = '\033[1;35m'[static]

string colors.Color.NEGRITO = '\033[1m'[static]

string colors.Color.PRETO = '\033[1;30m'[static]

string colors.Color.RESET = '\033[0m'[static]

string colors.Color.VERDE = '\033[92m'[static]

string colors.Color.VERDE_CLARO = '\033[1;92m'[static]

string colors.Color.VERMELHO = '\033[91m'[static]
```

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

pyFirewall-v1.1.1/colors.py

Referência à classe ipv4.logica_ipv4.DeleteRegra

Atributos Públicos Estáticos

```
delete_INPUT = LogicasMenu1("sudo iptables -D INPUT {}")
delete_FORWARD = LogicasMenu1("sudo iptables -D FORWARD {}")
delete_OUTPUT = LogicasMenu1("sudo iptables -D OUTPUT {}")
```

Descrição detalhada

Aqui os comandos do iptables para deletar regras.

Documentação dos dados membro

```
ipv4.logica_ipv4.DeleteRegra.delete_FORWARD = LogicasMenu1("sudo iptables -D FORWARD {}")[static]
```

```
ipv4.logica_ipv4.DeleteRegra.delete_INPUT = LogicasMenu1("sudo iptables -D INPUT {}")[static]
```

```
ipv4.logica_ipv4.DeleteRegra.delete_OUTPUT = LogicasMenu1("sudo iptables -D OUTPUT {}")[static]
```

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

pyFirewall-v1.1.1/ipv4/logica_ipv4.py

Referência à classe ipv4.logica_ipv4.IpRegras

Atributos Públicos Estáticos

```
ip_ACCEPT_tab_INPUT = LogicasMenu1("sudo iptables -A INPUT -s {} -j ACCEPT")
ip_DROP_tab_INPUT = LogicasMenu1("sudo iptables -A INPUT -s {} -j DROP")
ip_ACCEPT_tab_FORWARD = LogicasMenu1("sudo iptables -A FORWARD -s {} -j ACCEPT")
ip_DROP_tab_FORWARD = LogicasMenu1("sudo iptables -A FORWARD -s {} -j DROP")
ip_ACCEPT_tab_OUTPUT = LogicasMenu1("sudo iptables -A OUTPUT -s {} -j ACCEPT")
ip_DROP_tab_OUTPUT = LogicasMenu1("sudo iptables -A OUTPUT -s {} -j DROP")
```

Descrição detalhada

Aqui as regras especificas para ip

Documentação dos dados membro

```
ipv4.logica_ipv4.IpRegras.ip_ACCEPT_tab_FORWARD = LogicasMenu1("sudo iptables -A FORWARD -s {} -j ACCEPT")[static]
```

```
ipv4.logica_ipv4.IpRegras.ip_ACCEPT_tab_INPUT = LogicasMenu1("sudo iptables -A INPUT -s {} -j ACCEPT")[static]
```

```
ipv4.logica_ipv4.IpRegras.ip_ACCEPT_tab_OUTPUT = LogicasMenu1("sudo iptables -A OUTPUT -s {} -j ACCEPT")[static]
```

```
ipv4.logica_ipv4.IpRegras.ip_DROP_tab_FORWARD = LogicasMenu1("sudo iptables -A FORWARD -s {} -j DROP")[static]
```

```
ipv4.logica_ipv4.IpRegras.ip_DROP_tab_INPUT = LogicasMenu1("sudo iptables -A INPUT -s {} -j DROP")[static]
```

```
ipv4.logica_ipv4.IpRegras.ip_DROP_tab_OUTPUT = LogicasMenu1("sudo iptables -A OUTPUT -s {} -j DROP")[static]
```

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

pyFirewall-v1.1.1/ipv4/logica_ipv4.py

Referência à classe ipv4.logica_ipv4.LogicasMenu1

Membros públicos

```
def __init__(self, command)
def start_command(self)
def delete_id(self)
def port_change(self)
def ip_func_regra(self)
```

Atributos Públicos

command

Descrição detalhada

Estes metodos executam comandos do os.system().

Documentação dos Construtores & Destrutor

```
def ipv4.logica_ipv4.LogicasMenu1.__init__( self,  command)
    21     def __init__(self, command):
    22         self.command = command
    23
```

Documentação dos métodos

def ipv4.logica_ipv4.LogicasMenu1.delete_id (self)

```
31     def delete_id(self):
32         id = int(input(Color.VERMELHO + " digite numero da regra a ser
deletada \n>>" + Color.RESET))
33         os.system(self.command.format(id))
34
35
```

def ipv4.logica_ipv4.LogicasMenu1.ip_func_regra (self)

```
43     def ip_func_regra(self):
44         ip = str(input(Color.VERMELHO + "Digite o ip Escolhido \n IP >> " +
Color.RESET))
45         os.system(self.command.format(ip))
46
```

def ipv4.logica_ipv4.LogicasMenu1.port_change (self)

```
37     def port_change(self):
38         port = str(input(Color.VERMELHO + "Digite a Porta Escolhida \n PORT
>> " + Color.RESET))
39         os.system(self.command.format(port))
40
41
```

def ipv4.logica_ipv4.LogicasMenu1.start_command (self)

```
26     def start_command(self):
27         os.system(self.command)
28
29
```

Documentação dos dados membro

ipv4.logica_ipv4.LogicasMenu1.command

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

pyFirewall-v1.1.1/ipv4/logica_ipv4.py

Referência à classe `ipv4.logica_ipv4.RegrasList`

Atributos Públicos Estáticos

```
ports_tab_input_accept = LogicasMenu1("sudo iptables -A INPUT -p tcp --dport {} -j ACCEPT")
ports_tab_forward_accept = LogicasMenu1("sudo iptables -A FORWARD -p tcp --dport {} -j
ACCEPT")
ports_tab_output_accept = LogicasMenu1("sudo iptables -A OUTPUT -p tcp --dport {} -j
ACCEPT")
ports_tab_input_drop = LogicasMenu1("sudo iptables -A INPUT -p tcp --dport {} -j DROP")
ports_tab_forward_drop = LogicasMenu1("sudo iptables -A FORWARD -p tcp --dport {} -j
DROP")
ports_tab_output_drop = LogicasMenu1("sudo iptables -A OUTPUT -p tcp --dport {} -j DROP")
```

Descrição detalhada

Aqui esta os comandos de iptables para regra de firewall,
para desbloqueio de portas.

Documentação dos dados membro

```
ipv4.logica_ipv4.RegrasList.ports_tab_forward_accept = LogicasMenu1("sudo iptables
-A FORWARD -p tcp --dport {} -j ACCEPT")[static]
```

```
ipv4.logica_ipv4.RegrasList.ports_tab_forward_drop = LogicasMenu1("sudo iptables -
A FORWARD -p tcp --dport {} -j DROP")[static]
```

```
ipv4.logica_ipv4.RegrasList.ports_tab_input_accept = LogicasMenu1("sudo iptables -A
INPUT -p tcp --dport {} -j ACCEPT")[static]
```

```
ipv4.logica_ipv4.RegrasList.ports_tab_input_drop = LogicasMenu1("sudo iptables -A
INPUT -p tcp --dport {} -j DROP")[static]
```

```
ipv4.logica_ipv4.RegrasList.ports_tab_output_accept = LogicasMenu1("sudo iptables -
A OUTPUT -p tcp --dport {} -j ACCEPT")[static]
```

```
ipv4.logica_ipv4.RegrasList.ports_tab_output_drop = LogicasMenu1("sudo iptables -A
OUTPUT -p tcp --dport {} -j DROP")[static]
```

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

`pyFirewall-v1.1.1/ipv4/logica_ipv4.py`

Referência à classe `ipv4.logica_ipv4.SaveTable`

Atributos Públicos Estáticos

`status_service = LogicasMenu1("sudo systemctl status netfilter-persistent.service")`

Descrição detalhada

Aqui os comandos iptables para ver o status do serviço iptables, restartar o serviço e iniciar o serviço.

Documentação dos dados membro

`ipv4.logica_ipv4.SaveTable.status_service = LogicasMenu1("sudo systemctl status netfilter-persistent.service")[static]`

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

`pyFirewall-v1.1.1/ipv4/logica_ipv4.py`

Documentação do ficheiro

Referência ao ficheiro pyFirewall-v1.1.1/banner.py

Namespaces

banner

Funções

def **banner.header_banner** ()

Referência ao ficheiro pyFirewall-v1.1.1/colors.py

Componentes

class `colors.Color`

Namespaces

`colors`

Referência ao ficheiro pyFirewall-v1.1.1/ipv4/__init__.py

Namespaces

`ipv4`

Referência ao ficheiro pyFirewall-v1.1.1/ipv6/__init__.py

Namespaces

`ipv6`

Referência ao ficheiro pyFirewall-v1.1.1/ipv4/logica_ipv4.py

Componentes

class `ipv4.logica_ipv4.LogicasMenu1`

class `ipv4.logica_ipv4.RegrasList`

class `ipv4.logica_ipv4.DeleteRegra`

class `ipv4.logica_ipv4.SaveTable`

class `ipv4.logica_ipv4.IpRegras`

Namespaces

`ipv4.logica_ipv4`

Referência ao ficheiro pyFirewall-v1.1.1/pyfirewall.py

Namespaces

pyfirewall

Funções

```
def pyfirewall.menu_main_ipv4 ()  
    INCIO DO BLOCO DE MENU IPV4 #####.
```

Variáveis

```
pyfirewall.ver_regras = LogicasMenu1("sudo iptables -L --line-numbers")  
pyfirewall.delete = LogicasMenu1("sudo iptables -D INPUT")
```

Referência ao ficheiro pyFirewall-v1.1.1/README.md