Documentação 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:

adiconado regras de bloqueios e liberação de iPs especificos
correção de bugs

v1.0.4:

alterações na interface do usuario

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 diminur 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:

Índice dos componentes

Lista de componentes

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

Índice dos ficheiros

Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

Documentação dos namespaces

Referência ao namespace banner

Funções

def **header_banner** ()

Descrição detalhada

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

Documentação das funções

def banner.header_banner ()

```
 9 def header_banner():
10       print(Color.AMARELO +
11                     '''
12
13
14                                                              ,,
,,    ,,
15                                        `7MM"""YMM    db
`7MM   7MM
16                                          MM     `7
MM    MM
17             `7MMpdMAo. `7M'    `MF'   MM    d    `7MM   `7Mb,od8   .gP"Ya   `7M'
,A     `MF' ,6"Yb.    MM    MM
18               MM    `Wb    VA    ,V     MM""MM       MM     MM' "' ,M'    Yb    VA
,VAA    ,V   8)    MM    MM    MM
19               MM     M8     VA ,V      MM    Y       MM     MM       8M"""""""
VA ,V   VA ,V     ,pm9MM    MM    MM
20               MM    ,AP      VVV       MM            MM     MM       YM.     ,
VVV     VVV     8M    MM    MM    MM
21               MMbmmd'       ,V      .JMML.       .JMML..JMML.     `Mbmmd'
W        W      `Moo9^Yo.JMML.JMML. v1.1.1
22               MM             ,V
23             .JMML.       OOb"
24
25                                              Copyright (c) 2022 Juan Carlos
Bindez
26
27                                                *[Ctrl + C] Para Sair do
Programa
28                     '''
29       + Color.RESET)
30
```

Referência ao namespace colors

Componentes

class **Color**

Descrição detalhada

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

Referência ao namespace ipv4

Namespaces

logica_ipv4

Referência ao namespace ipv4.logica_ipv4

Componentes

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

Descrição detalhada

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

Referência ao namespace ipv6

Descrição detalhada

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

Referência ao namespace pyfirewall

Funções

def **menu_main_ipv4** ()

INCIO DO BLOCO DE MENU IPV4 ####.

Variáveis

ver_regras = **LogicasMenu1**("sudo iptables -L --line-numbers")
delete = **LogicasMenu1**("sudo iptables -D INPUT")

Descrição detalhada

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

Documentação das funções

def pyfirewall.menu_main_ipv4 ()

INCIO DO BLOCO DE MENU IPV4 ####.

```
20      def menu_main_ipv4():
21
22
23          def Ver_regras_firewall():
24               # ve regras existentes no firewall
25               os.system("clear")
26               ver_regras.start_command()
27               menu_main_ipv4()
28
29
30          def deletar_regras_firewall():
31               os.system("clear")
32               ver_regras.start_command()
33               header_banner()
34               # deleta id de regra no firewall
35               print(Color.AMARELO +
36
37                    '''
38                                                  Deletar de qual tabela?
39
40                                          *[0]Voltar
41                                          *[1]INPUT
42                                          *[2]FORWARD
43                                          *[3]OUTPUT
44
45                    '''
46               + Color.RESET)
47
48               choice_delete = str(input(">>"))
49
50               if choice_delete == "0":
51                    os.system("clear")
52                    menu_main_ipv4()
53
54               elif choice_delete == "1":
55                    DeleteRegra.delete_INPUT.delete_id()
56                    os.system("clear")
57                    ver_regras.start_command()
```

```
58                    menu_main_ipv4()
59
60               elif choice_delete == "2":
61                    DeleteRegra.delete_FORWARD.delete_id()
62                    os.system("clear")
63                    ver_regras.start_command()
64                    menu_main_ipv4()
65
66               elif choice_delete == "3":
67                    DeleteRegra.delete_OUTPUT.delete_id()
68                    os.system("clear")
69                    ver_regras.start_command()
70                    menu_main_ipv4()
71
72               else:
73                    os.system("clear")
74                    print("ops, digite apenas os numeros listados!")
75                    menu_main_ipv4()
76
77               os.system("clear")
78               ver_regras.start_command()
79               delete.delete_id()
80               menu_main_ipv4()
81
82
83
84          def regras_de_ports_firewall():
85
86
87               def regra_port_INPUT():
88                    header_regra_port()
89                    choice_regra = str(input(">>"))
90
91                    if choice_regra == "0":
92                         os.system("clear")
93                         menu_main_ipv4()
94
95                    elif choice_regra == "1":
96                         RegrasList.ports_tab_input_accept.port_change()
97                         os.system("clear")
98                         ver_regras.start_command()
99                         menu_main_ipv4()
100
101                   elif choice_regra == "2":
102                        RegrasList.ports_tab_input_drop.port_change()
103                        os.system("clear")
104                        ver_regras.start_command()
105                        menu_main_ipv4()
106
107                   else:
108                        os.system("clear")
109                        print("ops, digite apenas os numeros listados!")
110                        menu_main_ipv4()
111
112
113              def header_regra_port():
114                   os.system("clear")
115                   header_banner()
116                   print(Color.AMARELO +
117                   '''
118                                        *[0]Voltar
119                                        *[1]ACCEPT
120                                        *[2]DROP
121
122                   '''
123                   + Color.RESET)
124
125
126
127              def regra_port_FORWARD():
128                   header_regra_port()
129                   choice_regra = str(input(">>"))
130
131                   if choice_regra == "1":
132                        RegrasList.ports_tab_forward_accept.port_change()
133                        os.system("clear")
134                        ver_regras.start_command()
```

```
135                         menu_main_ipv4()
136
137                    elif choice_regra == "2":
138                         RegrasList.ports_tab_forward_drop.port_change()
139                         os.system("clear")
140                         ver_regras.start_command()
141                         menu_main_ipv4()
142
143                    else:
144                         os.system("clear")
145                         print("ops, digite apenas os numeros listados!")
146                         menu_main_ipv4()
147
148
149               def regra_port_OUTPUT():
150                    header_regra_port()
151                    choice_regra = str(input(">>"))
152
153                    if choice_regra == "1":
154                         RegrasList.ports_tab_output_accept.port_change()
155                         os.system("clear")
156                         ver_regras.start_command()
157                         menu_main_ipv4()
158
159                    elif choice_regra == "2":
160                         RegrasList.ports_tab_output_drop.port_change()
161                         os.system("clear")
162                         ver_regras.start_command()
163                         menu_main_ipv4()
164
165                    else:
166                         os.system("clear")
167                         print("ops, digite apenas os numeros listados!")
168                         menu_main_ipv4()
169
170
171
172               os.system("clear")
173               ver_regras.start_command()
174               header_banner()
175               print(Color.AMARELO +
176                    '''
177                                                  Escolha a Tabela
178
179                                        *[0]Voltar
180                                        *[1]INPUT
181                                        *[2]FORWARD
182                                        *[3]OUTPUT
183
184                    '''
185               + Color.RESET)
186
187               choice_tab = str(input(">>"))
188
189               if choice_tab == "0":
190                    os.system("clear")
191                    menu_main_ipv4()
192
193               elif choice_tab == "1":
194                    regra_port_INPUT()
195
196               elif choice_tab == "2":
197                    regra_port_FORWARD()
198
199               elif choice_tab == "3":
200                    regra_port_OUTPUT()
201
203
204
205
206          def salva_regras_firewall():
207               os.system("sudo service netfilter-persistent save")
208               time.sleep(2)
209               os.system("sudo systemctl restart netfilter-
persistent.service")
210               time.sleep(2)
211               os.system("sudo systemctl status netfilter-persistent.service")
```

```
212                 os.system("clear")
213                 menu_main_ipv4()
214
215
216           def netfilter_install():
217                 os.system("sudo apt-get install netfilter-persistent.service")
218                 os.system("sudo apt-get install iptables-persistent")
219                 time.sleep(2)
220                 os.system("clear")
221                 menu_main_ipv4()
222
223
224           def exclui_tab_firewall():
225                 os.system("clear")
226                 ver_regras.start_command()
227                 header_banner()
228
229                 print(Color.AMARELO +
230
231                       '''
232                                                 Escolha a Tabela a ser
Excluída
233
234                                           *[0]Voltar
235                                           *[1]INPUT
236                                           *[2]FORWARD
237                                           *[3]OUTPUT
238                                           *[4]Todas as tabelas
239
240                       '''
241                 + Color.RESET)
242
243                 escolha = str(input(">>"))
244
245                 if escolha == "0":
246                       os.system("clear")
247                       menu_main_ipv4()
248
249                 elif escolha == "1":
250                       os.system("sudo iptables -F INPUT")
251                       os.system("clear")
252                       ver_regras.start_command()
253                       menu_main_ipv4()
254
255                 elif escolha == "2":
256                       os.system("sudo iptables -F FORWARD")
257                       os.system("clear")
258                       ver_regras.start_command()
259                       menu_main_ipv4()
260
261                 elif escolha == "3":
262                       os.system("sudo iptables -F OUTPUT")
263                       os.system("clear")
264                       ver_regras.start_command()
265                       menu_main_ipv4()
266
267                 elif escolha == "4":
268                       os.system("sudo iptables -F")
269                       os.system("clear")
270                       ver_regras.start_command()
271                       menu_main_ipv4()
272
273                 else:
274                       os.system("clear")
275                       print("ops, digite apenas os numeros listados!")
276                       menu_main_ipv4()
277
278
279
280
281           def ip_regras():
282
283                 def header_escolha7():
284                       os.system("clear")
285                       header_banner()
286
287                       print(Color.AMARELO +
```

```
                    '''
                                        *[0]Voltar
                                        *[1]ACCEPT
                                        *[2]DROP


                    '''
          + Color.RESET)


def ip_regra_INPUT_ACCEPT():
     IpRegras.ip_ACCEPT_tab_INPUT.ip_func_regra()
     os.system("clear")
     ver_regras.start_command()
     menu_main_ipv4()


def ip_regra_FORWARD_ACCEPT():
     IpRegras.ip_ACCEPT_tab_FORWARD.ip_func_regra()
     os.system("clear")
     ver_regras.start_command()
     menu_main_ipv4()



def ip_regra_OUTPUT_ACCEPT():
     IpRegras.ip_ACCEPT_tab_OUTPUT.ip_func_regra()
     os.system("clear")
     ver_regras.start_command()
     menu_main_ipv4()



def ip_regra_INPUT_DROP():
     IpRegras.ip_DROP_tab_INPUT.ip_func_regra()
     os.system("clear")
     ver_regras.start_command()
     menu_main_ipv4()


def ip_regra_FORWARD_DROP():
     IpRegras.ip_DROP_tab_FORWARD.ip_func_regra()
     os.system("clear")
     ver_regras.start_command()
     menu_main_ipv4()


def ip_regra_OUTPUT_DROP():
     IpRegras.ip_DROP_tab_OUTPUT.ip_func_regra()
     os.system("clear")
     ver_regras.start_command()
     menu_main_ipv4()


os.system("clear")
header_banner()
print(Color.AMARELO +
     '''
                                          Escolha a Tabela

                              *[0]Voltar
                              *[1]INPUT
                              *[2]FORWARD
                              *[3]OUTPUT
     '''
+ Color.RESET)

escolha7 = str(input(">>"))

if escolha7 == "0":
     os.system("clear")
     menu_main_ipv4()

elif escolha7 == "1":
     header_escolha7()
     escolha = str(input(">>"))
```

```
365
366                     if escolha == "0":
367                          os.system("clear")
368                          menu_main_ipv4()
369
370                     elif escolha == "1":
371                          ip_regra_INPUT_ACCEPT()
372
373                     elif escolha == "2":
374                          ip_regra_INPUT_DROP()
375
376
377                elif escolha7 == "2":
378                     header_escolha7()
379                     escolha = str(input(">>"))
380
381                     if escolha == "0":
382                          os.system("clear")
383                          menu_main_ipv4()
384
385                     elif escolha == "1":
386                          ip_regra_FORWARD_ACCEPT()
387
388                     elif escolha == "2":
389                          ip_regra_FORWARD_DROP()
390
391                elif escolha7 == "3":
392                     header_escolha7()
393                     escolha = str(input(">>"))
394
395                     if escolha == "0":
396                          os.system("clear")
397                          menu_main_ipv4()
398
399                     elif escolha == "1":
400                          ip_regra_OUTPUT_ACCEPT()
401
402                     elif escolha == "2":
403                          ip_regra_OUTPUT_DROP()
404
405                else:
406                     os.system("clear")
407                     print("Ops, Digite apenas os numeros listados!")
408                     ip_regras()
409
410
411
412
414
415
416
417
418           header_banner()
419           print(Color.AMARELO +
420                '''
421                                          *[1]Ver regras
422                                          *[2]Delete regra
423                                          *[3]Ports
424                                          *[4]Salvar
425                                          *[5]Instalar o netfilter-
persistent.service
426                                          *[6]Excluir tabelas
427                                          *[7]Ip (regras para IPs especificos)
428
429                '''
430           + Color.RESET)
431
432           choice = str(input(">>"))
433
434           if choice == "1":
435                Ver_regras_firewall()
436
437           elif choice == "2":
438              deletar_regras_firewall()
439
440           elif choice == "3":
441                regras_de_ports_firewall()
```

```
442
443           elif choice == "4":
444                salva_regras_firewall()
445
446           elif choice == "5":
447                netfilter_install()
448
449           elif choice == "6":
450                exclui_tab_firewall()
451
452           elif choice == "7":
453                ip_regras()
454
455           else:
456                os.system("clear")
457                print("Digite Apenas os Números Listados!")
458                menu_main_ipv4()
459
460
461
```

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 {}")

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")

Documentação dos dados membro

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

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

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

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

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

ipv4.logica_ipv4.lpRegras.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

```
Executa Comandos.
```

Documentação dos Construtores & Destrutor

def ipv4.logica_ipv4.LogicasMenu1.__init__ (*self*, *command*)

```
15      def __init__(self, command):
16           self.command = command
17
```

Documentação dos métodos

def ipv4.logica_ipv4.LogicasMenu1.delete_id (*self*)

```
21      def delete_id(self):
22           id = int(input(Color.VERMELHO + " digite numero da regra a ser
deletada \n>>" + Color.RESET))
23           os.system(self.command.format(id))
24
```

def ipv4.logica_ipv4.LogicasMenu1.ip_func_regra (*self*)

```
29      def ip_func_regra(self):
30           ip = str(input(Color.VERMELHO + "Digite o ip Escolhido \n PORT >> " +
Color.RESET))
31           os.system(self.command.format(ip))
32
```

def ipv4.logica_ipv4.LogicasMenu1.port_change (*self*)

```
25      def port_change(self):
26           port = str(input(Color.VERMELHO + "Digite a Porta Escolhida \n PORT
>> " + Color.RESET))
27           os.system(self.command.format(port))
28
```

def ipv4.logica_ipv4.LogicasMenu1.start_command (*self*)

```
18      def start_command(self):
19           os.system(self.command)
20
```

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

ACCEPTS

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")

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