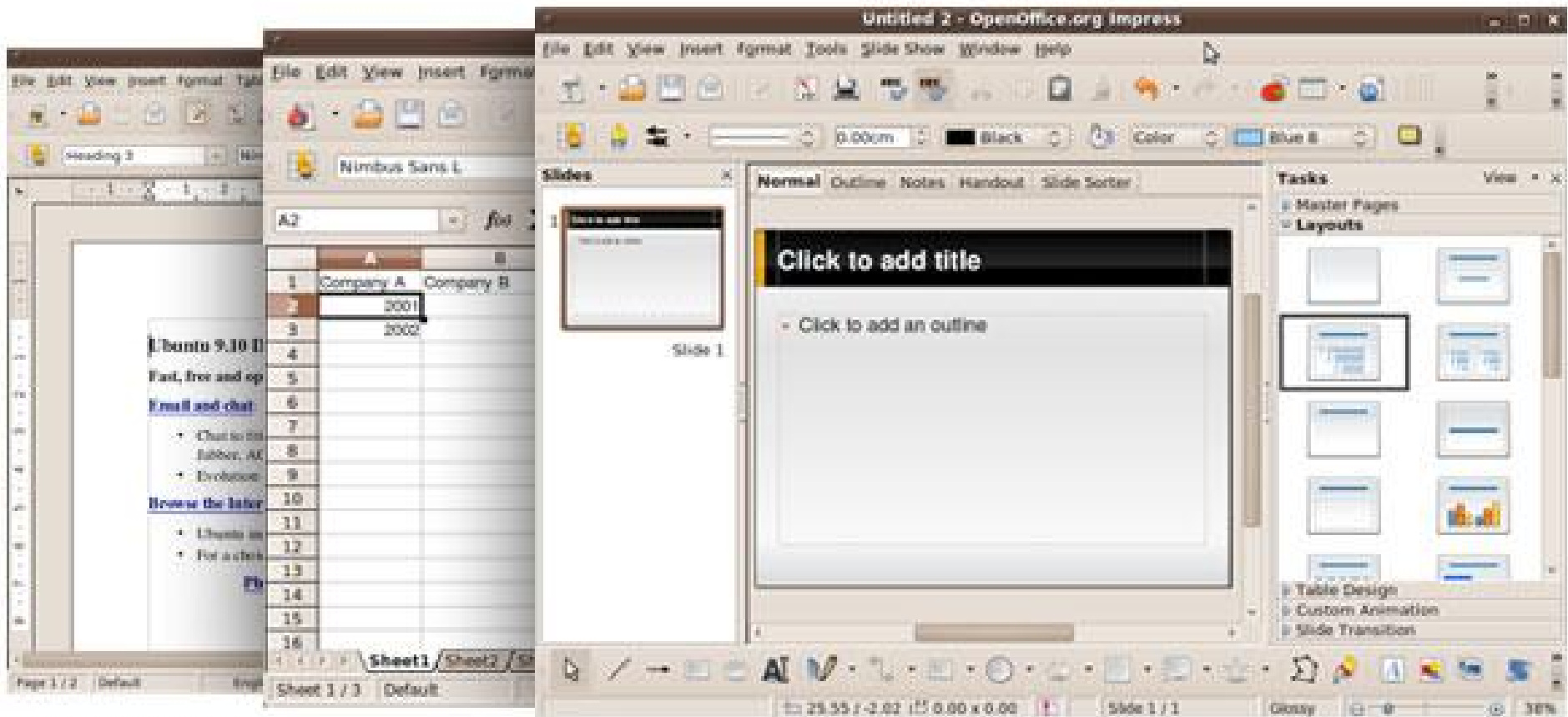


Iptables

Entendendo como fazer um firewall pessoal



1UP=2

0

ARMS BOMB
∞ 10

60

INSERT COIN



- AdvanceMAME
- Input (general)
- Input (this Game)
- Dip Switches
- Calibrate Joysticks
- Bookkeeping Info
- Game Information
- Game History
- Memory Card
- Video
- Audio
- Reset Game
- Return to Game

LEVEL-4

CREDIT 00

```
3 (% from "entry.html" import render_entry %)
4 (% block contents %)
5 (%- for post in posts %)
6   {{ render_entry(post) }}
7 (%- else %)
8   <h2>{% trans %}Welcome{% endtrans %}</h2>
9   <div class="notification">
10     <p>{% trans %}So far there are no entries in this blog.{% endtrans %}</p>
11   </div>
12 (%- endfor %)
13 (%- if pagination.necessary %)
14   <div class="pagination">
15     {{ pagination.generate() }}
16   </div>
17 (%- endif %)
18 (% endblock %)
```

```
[10,36 All] ~/Development/zine/zine/templates/index.html (unix)
425 May be overridden.
426
427 """
428 self.socket.close()
429
430 def fileno(self):
431     """Return socket file number.
432
433     Interface required by select().
434
435     """
436     return self.socket.fileno()
437
438 def get_request(self):
439     """Get the request and client address from the socket.
440
441     May be overridden.
442
443     """
444     return self.socket.accept()
445
446 def close_request(self, request):
447     """Called to clean up an individual request."""
448     request.close()
449
450
451 class UDPServer(TCPServer):
452     """UDP server class."""
453
454     allow_reuse_address = False
455
456     socket_type = socket.SOCK_DGRAM
457
458     max_packet_size = 8192
```

[448,23 65%] ~/Development/python-trunk/Lib/SocketServer.py (unix)

```
227 """A WSGI server that does threading."""
228 multithread = True
229
230
231 class ForkingWSGIServer(ForkingMixIn, BaseWSGIServer):
232     """A WSGI server that does forking."""
233     multiprocess = True
234
235     def __init__(self, host, port, app, processes=40, handler=None,
236                 passthrough_errors=False):
237         BaseWSGIServer.__init__(self, host, port, app, handler,
238                                 passthrough_errors)
239         self.max_children = processes
240
241
242 def make_server(host, port, app=None, threaded=False, processes=1,
243               request_handler=None, passthrough_errors=False):
244     """Create a new server instance that is either threaded, or forks
245     or just processes one request after another.
246     """
247     if threaded and processes > 1:
248         raise ValueError("cannot have a multithreaded and "
249                          "multi process server.")
250     elif threaded:
251         return ThreadedWSGIServer(host, port, app, request_handler,
252                                   passthrough_errors)
253     elif processes > 1:
254         return ForkingWSGIServer(host, port, app, processes, request_handler,
255                                  passthrough_errors)
256     else:
257         return BaseWSGIServer(host, port, app, request_handler,
258                               passthrough_errors)
259
260
261 def reloader_loop(extra_files=None, interval=1):
262     """When this function is run from the main thread, it will force other
263     threads to exit when any modules currently loaded change.
264
265     Copyright notice. This function is based on the autoreload.py from
266     the CherryPy trac which originated from WSGIKit which is now dead.
267
268     :param extra_files: a list of additional files it should watch.
269     """
270     def iter_module_files():
271         for module in sys.modules.values():
272             filename = getattr(module, '__file__', None)
273             if filename:
274                 while not os.path.isfile(filename):
275                     filename = os.path.dirname(filename)
276                 if not filename:
277                     break
278             else:
279                 if filename[-4:] in ('.pyc', '.pyo'):
280                     filename = filename[:-1]
```

[265,25 66%] serving.py (unix)

:e

__init__.py	contrib/	debug/	http.py	routing.py	serving.py	test.py	useragents.py	wrappers.py
_internal.py	datastructures.py	exceptions.py	local.py	script.py	templates.py	testapp.py	utils.py	
:e datastructures.py								

**Você tem
twitter?**





Você tem
Twitter?

**Será que estou
protegido?**

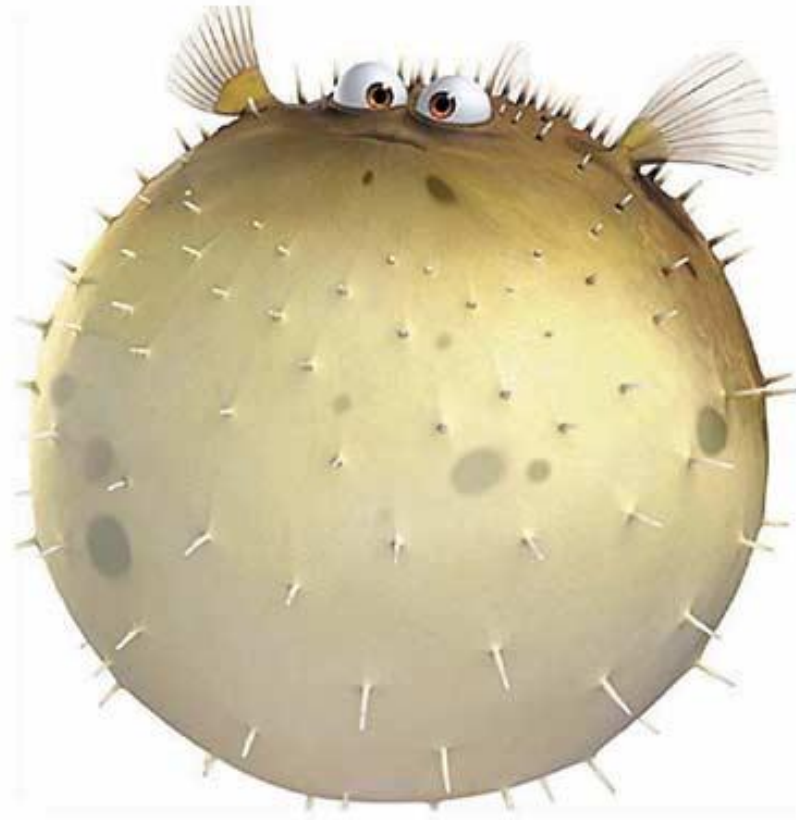


Port Scanner



nmap

Superscan



OpenSSH



PostgreSQL




APACHE CHIEF



SysAdmin



Filtro de Pacotes

A man in a black uniform and cap, holding a rifle, is shown in a dark setting. He has a speech bubble next to him that says "Tô de olho!".

**Tô de
olho!**

Vaza!



Motivos



Controle



O Yes

O No

Acesso Negado



2

Segurança





Observador



Vigilância



./comofás ?



Calma!
Não é tão difícil assim!

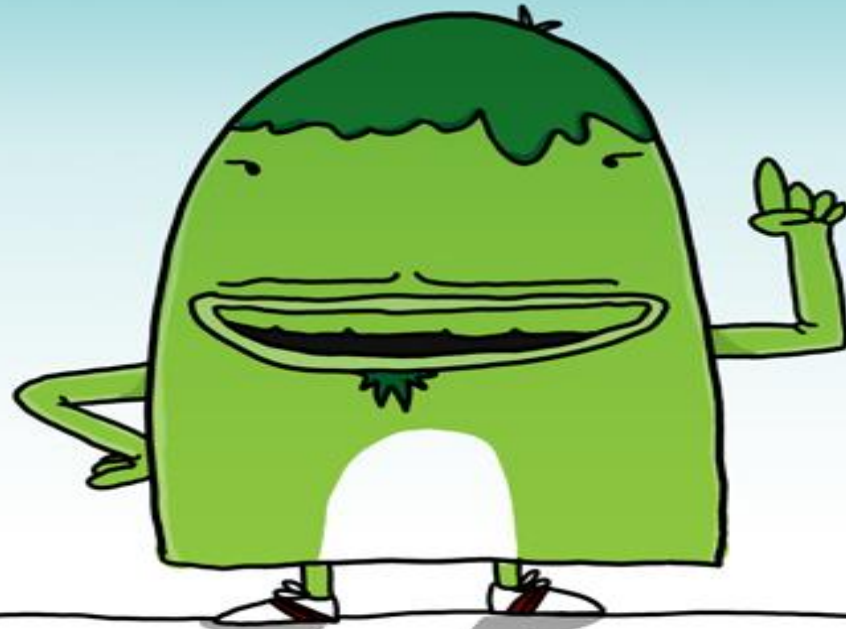
**Kernel com
suporte**

Iptables
(claro!)



Shell

LET'S START!



Objetivo



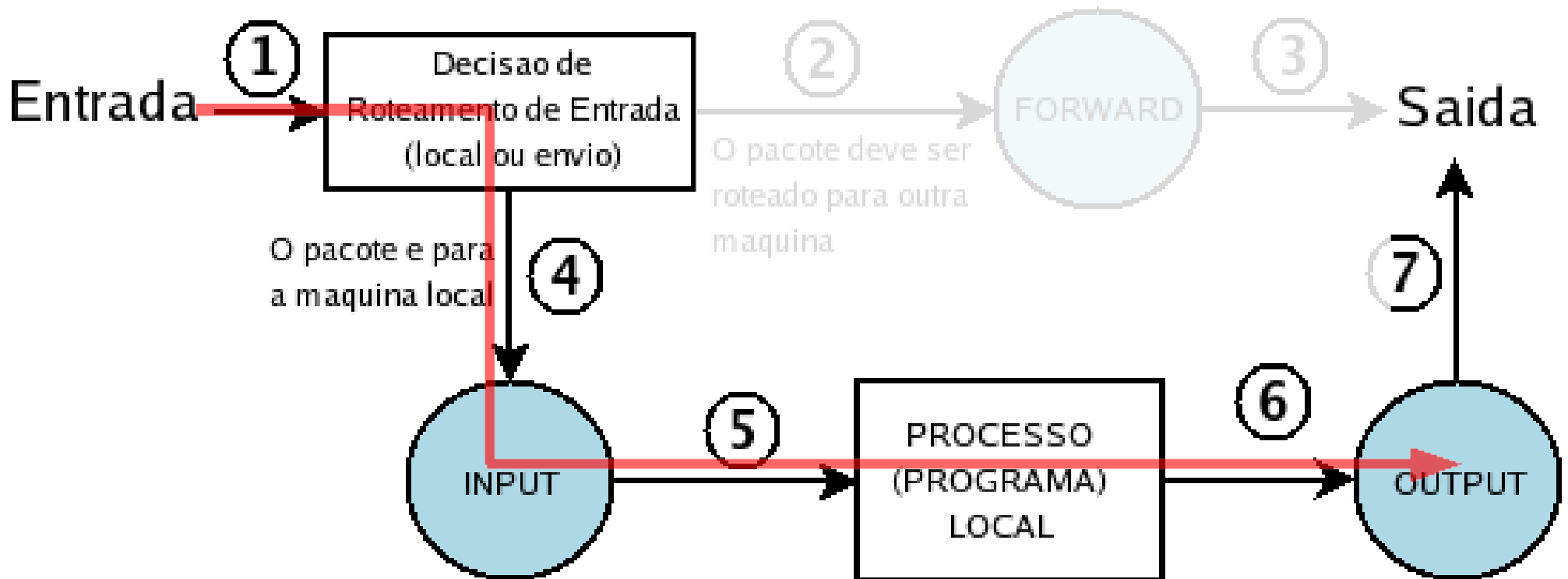
**Conexão PPP e ninguém
bisbilhotando**

1 - Bloquear tudo que vem de fora.

2 - Liberar serviços (se desejar)

3 - Fazer log do restante

Caminho do pacote



- 1. Sequência de Regras**
- 2. Análise e comparação**
- 3. Define o destino do pacote**
- 4. Análise da política da **chain****

Nosso foco

Chain INPUT



**É EXPRESSAMENTE
PROIBIDA**

**A ENTRADA DE PESSOAS
NÃO AUTORIZADAS.**

**Definir política
da chain**



Não vai subir ninguém!

iptables -P INPUT DROP

ENTRADA

PROIBIDA

Liberal Adicionar de regra al

iptables -A INPUT -i lo -j ACCEPT

Pacotes relacionados

iptables -A INPUT -m state
--state RELATED, ESTABLISHED
-j ACCEPT



**Liberrar serviços
necessários**

Liberando SSH

Adicionar Regra

```
iptables -A INPUT -p tcp -dport 22  
-j ACCEPT
```



Serviços e Portas

cat /etc/services



**Guardar logs de
bloqueios**

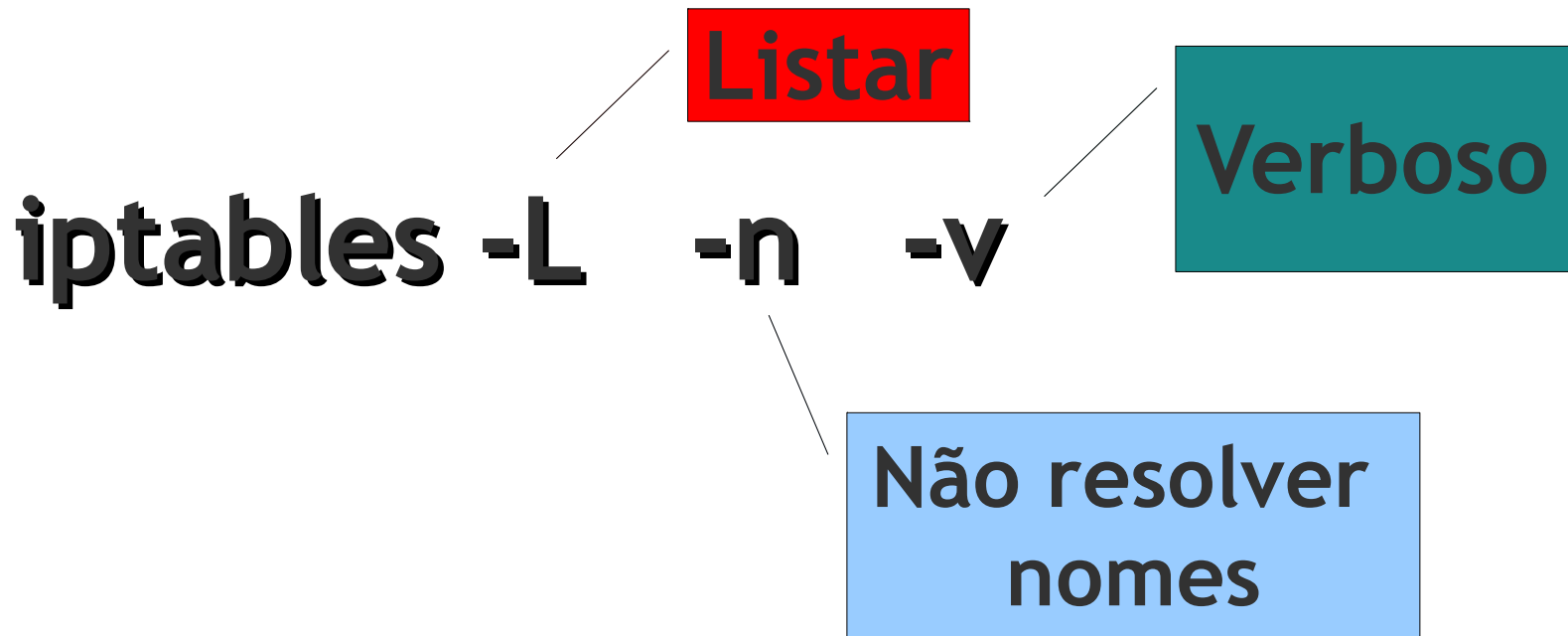
Log, log, log...

```
iptables -A INPUT -m limit --limit  
1/m -j LOG --log-prefix "INPUT: "
```

Olhando o log

```
tail -f /var/log/messages
```

Ver as regras ativas





I SEE WHAT YOU DID THERE.

E se eu fizer besteira?

```
iptables -D INPUT -p tcp -dport 22  
-j ACCEPT
```

```
iptables -D 1
```

Desfazer “tuto”!

Apagar todas as regras

```
iptables -F INPUT
```

```
iptables -P INPUT ACCEPT
```

Define a política

E se eu quiser mais?

<http://www.netfilter.org/documentation/>

**Valeu
galera!**





<http://www.almirmendes.net>



m3nd3s@gmail.com



@m3nd3s