Making USB Great Again with USBFILTER

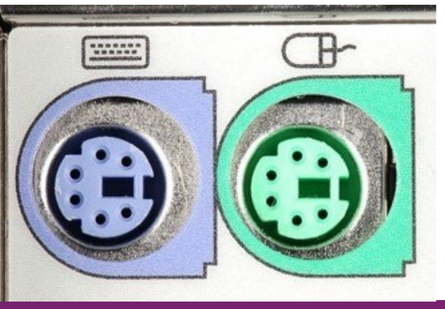
Dave Tian
Nolen Scaife
Kevin Butler
Patrick Traynor
University of Florida

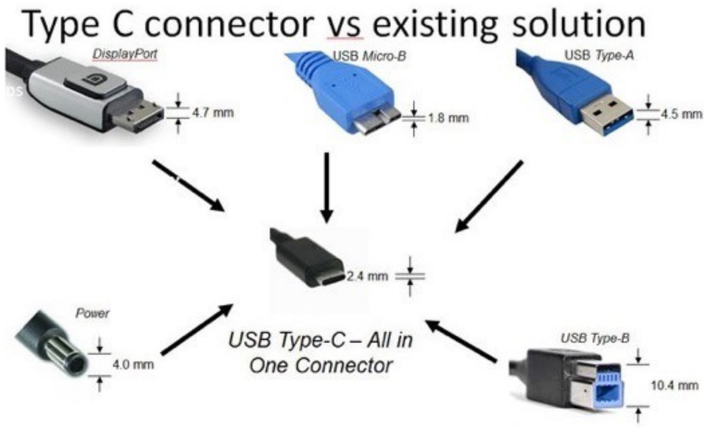
Adam Bates
University of Ilinois

Tradução:
Leandro Fabian Junior
Universidade Tecnológica Federal do Paraná

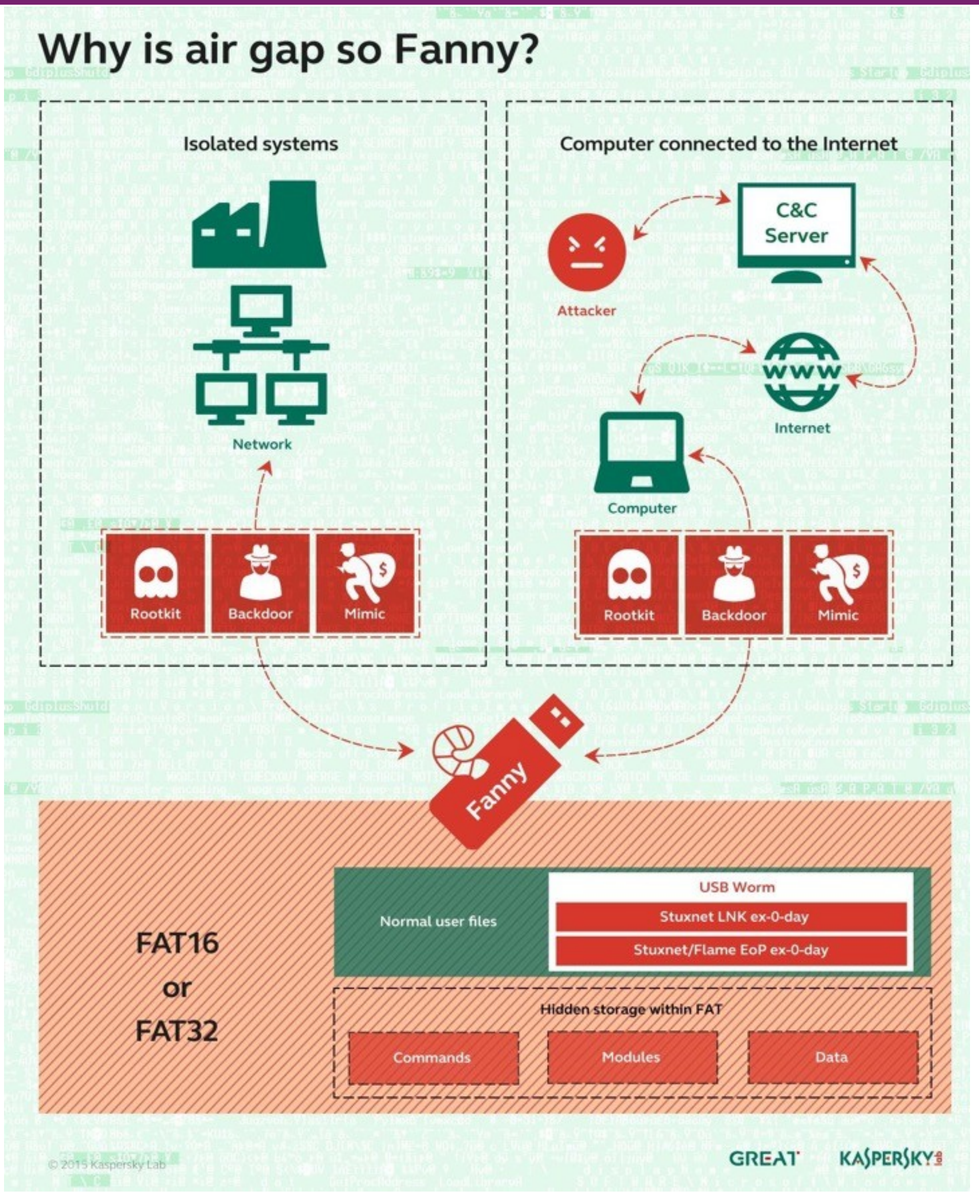
Por que o USB era ótimo?

- Universal Serial Bus
  - USB 1.0/2.0/3.0/3.1/Type-C
- Velocidade
  - 1.5 Mb/s → 10 Gb/s
- Ubíquo

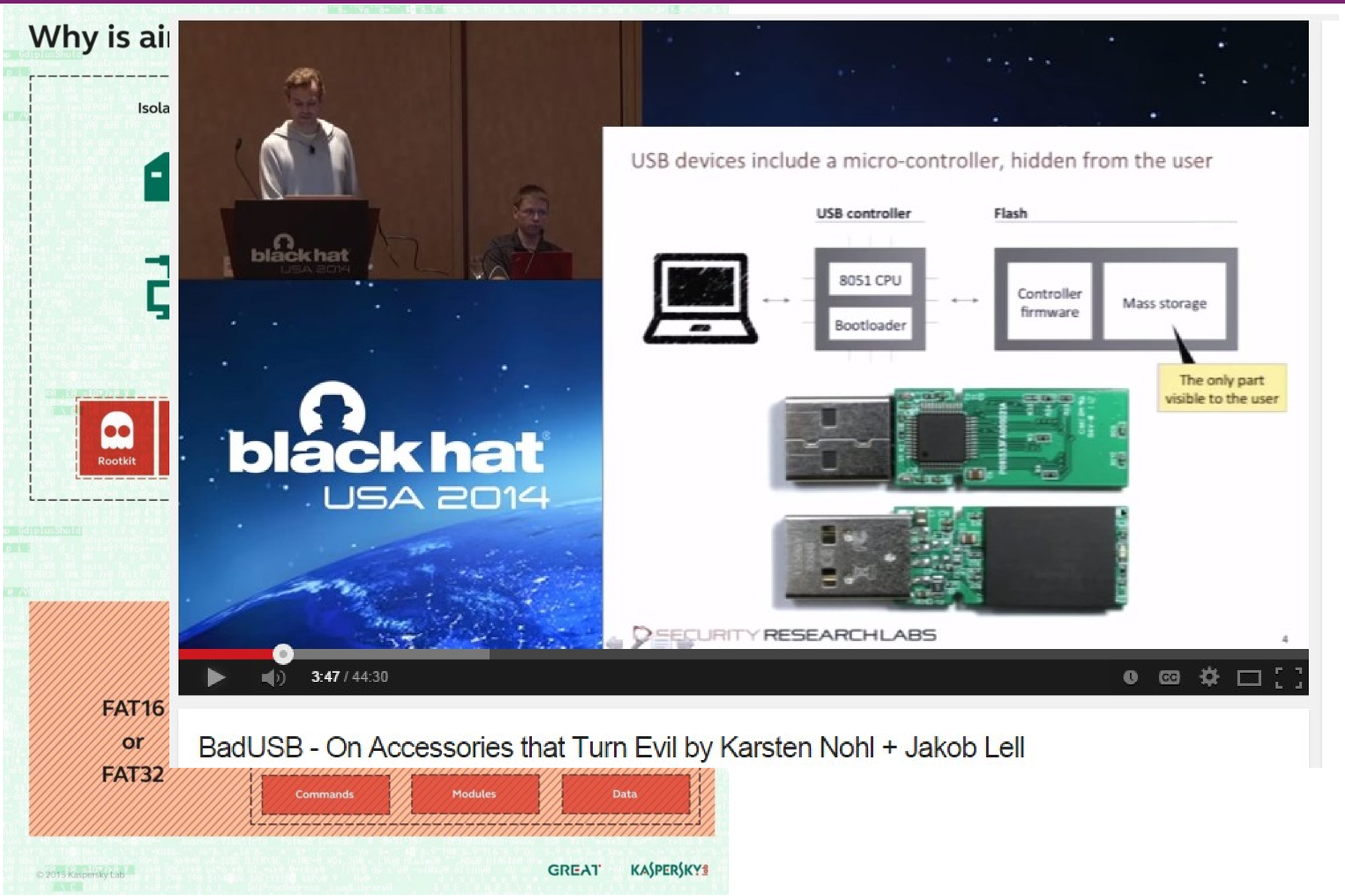




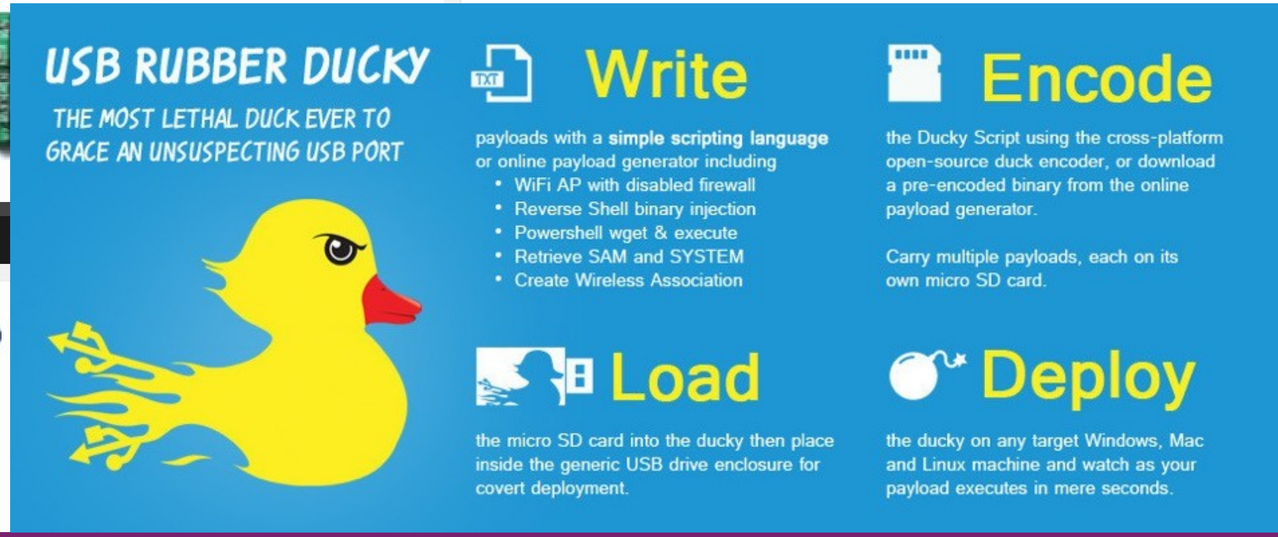
Por que o USB não é mais tão bom?



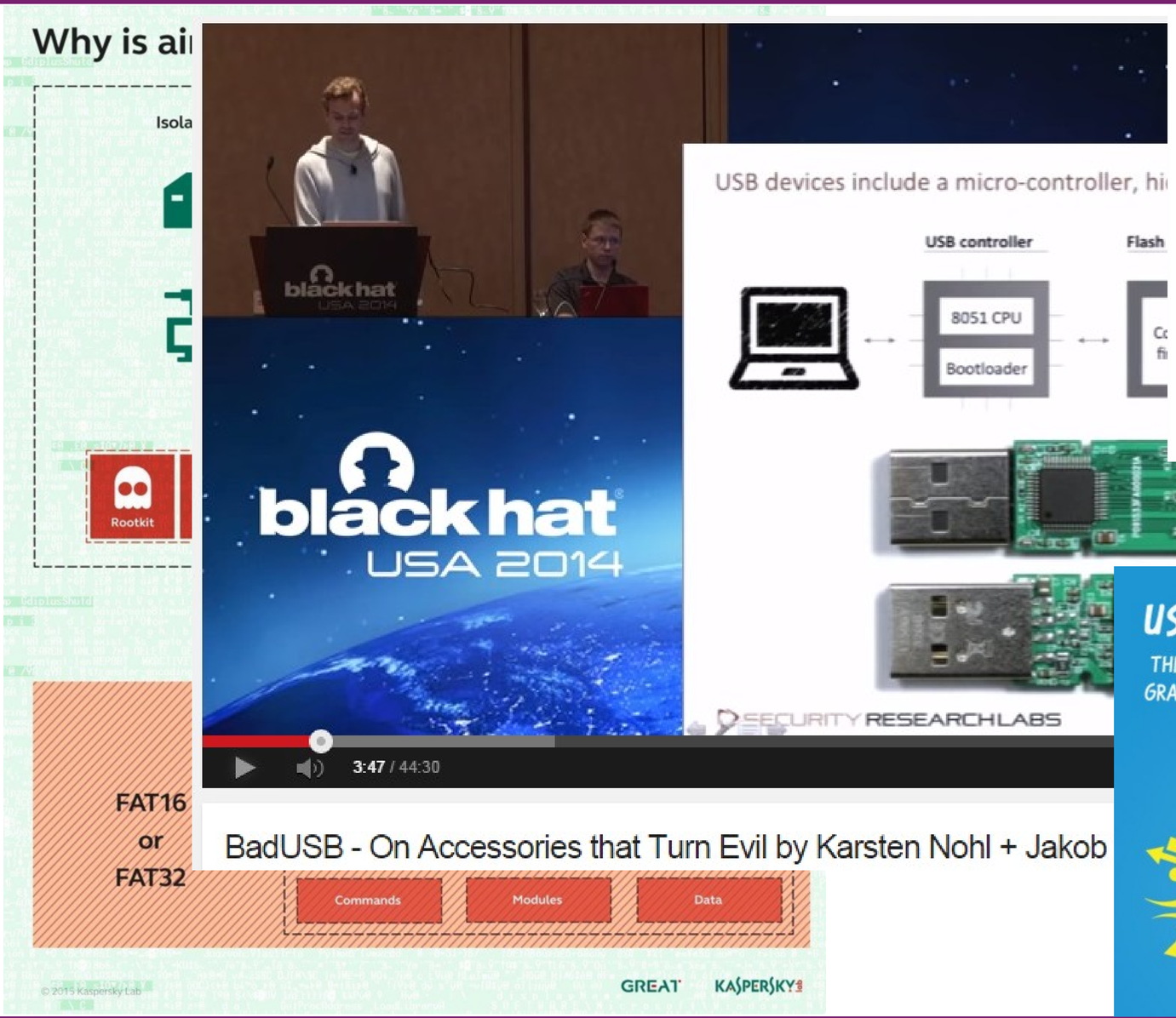
Por que o USB não é mais tão bom?

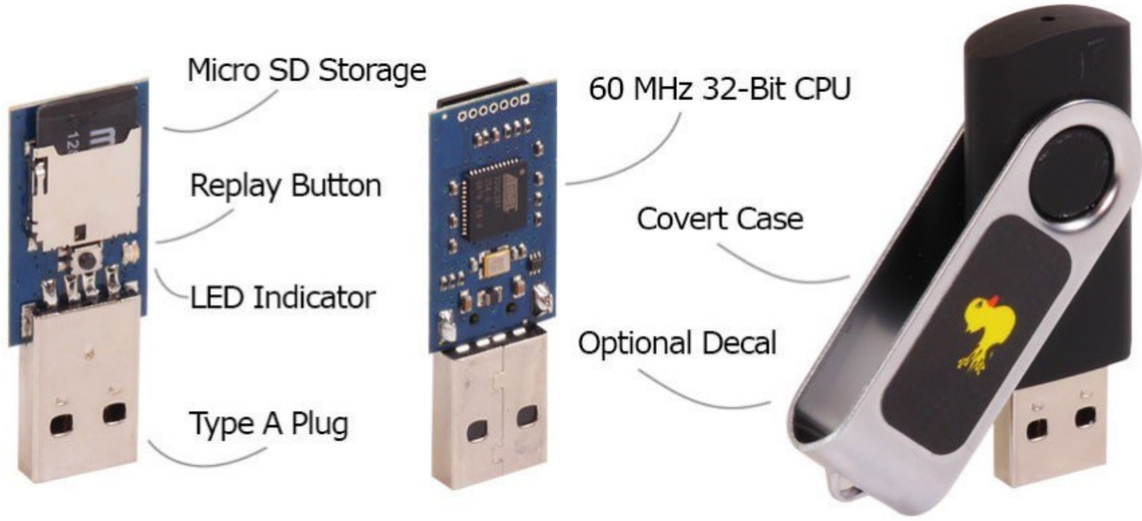


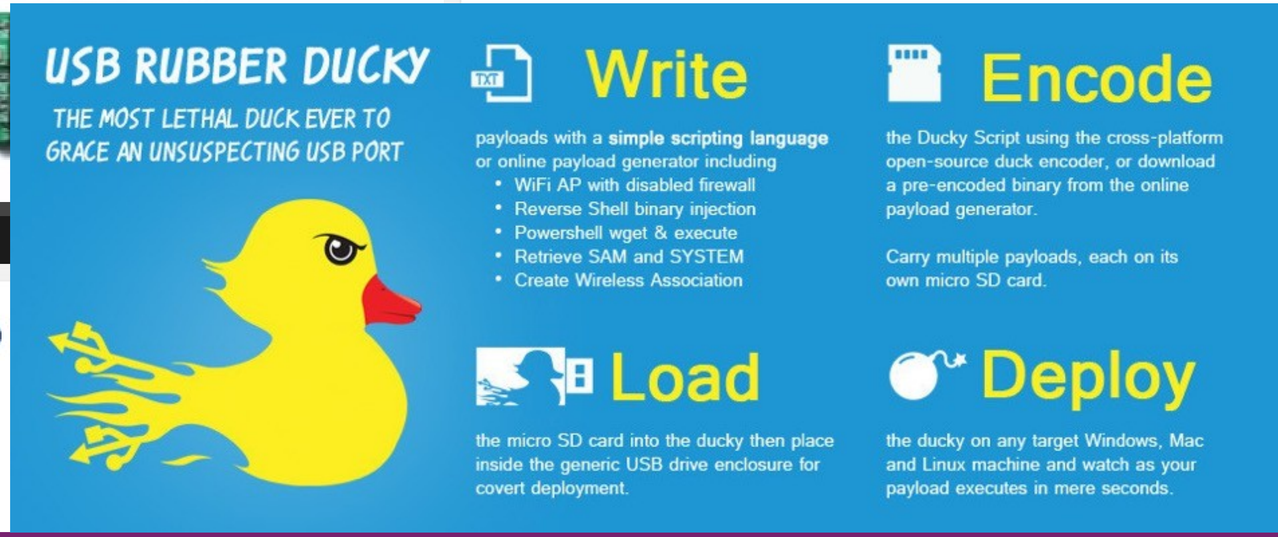
Por que o USB não é mais tão bom?



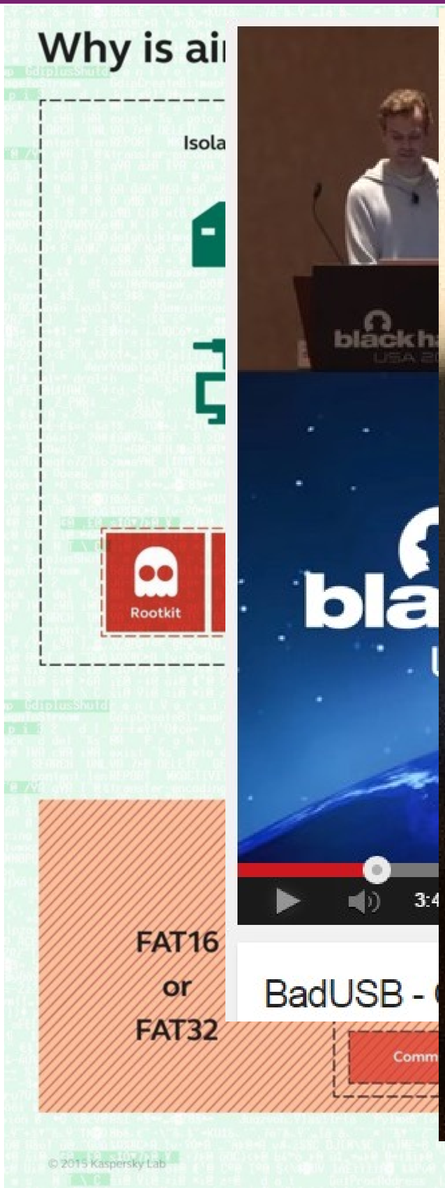
Por que o USB não é mais tão bom?



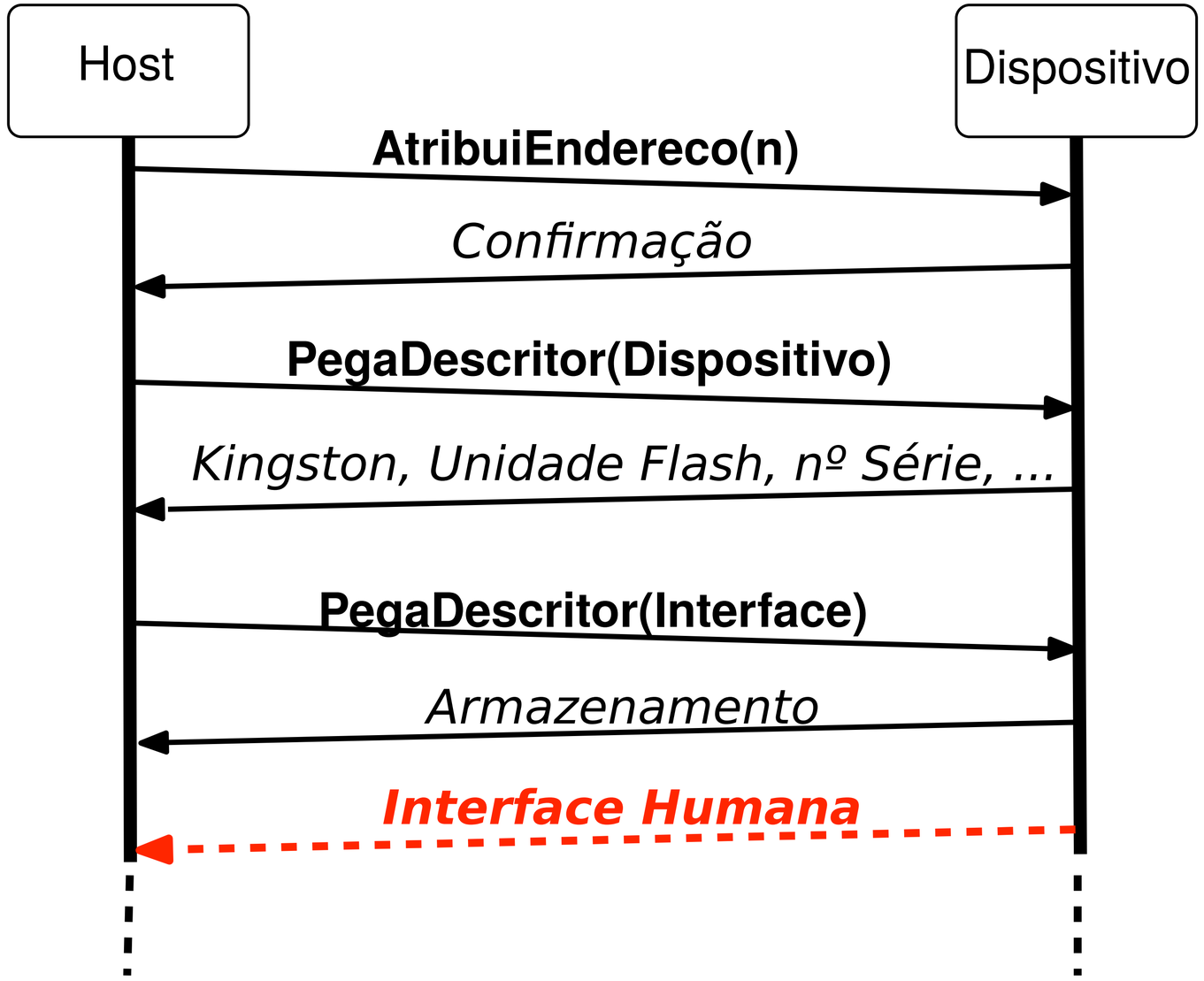




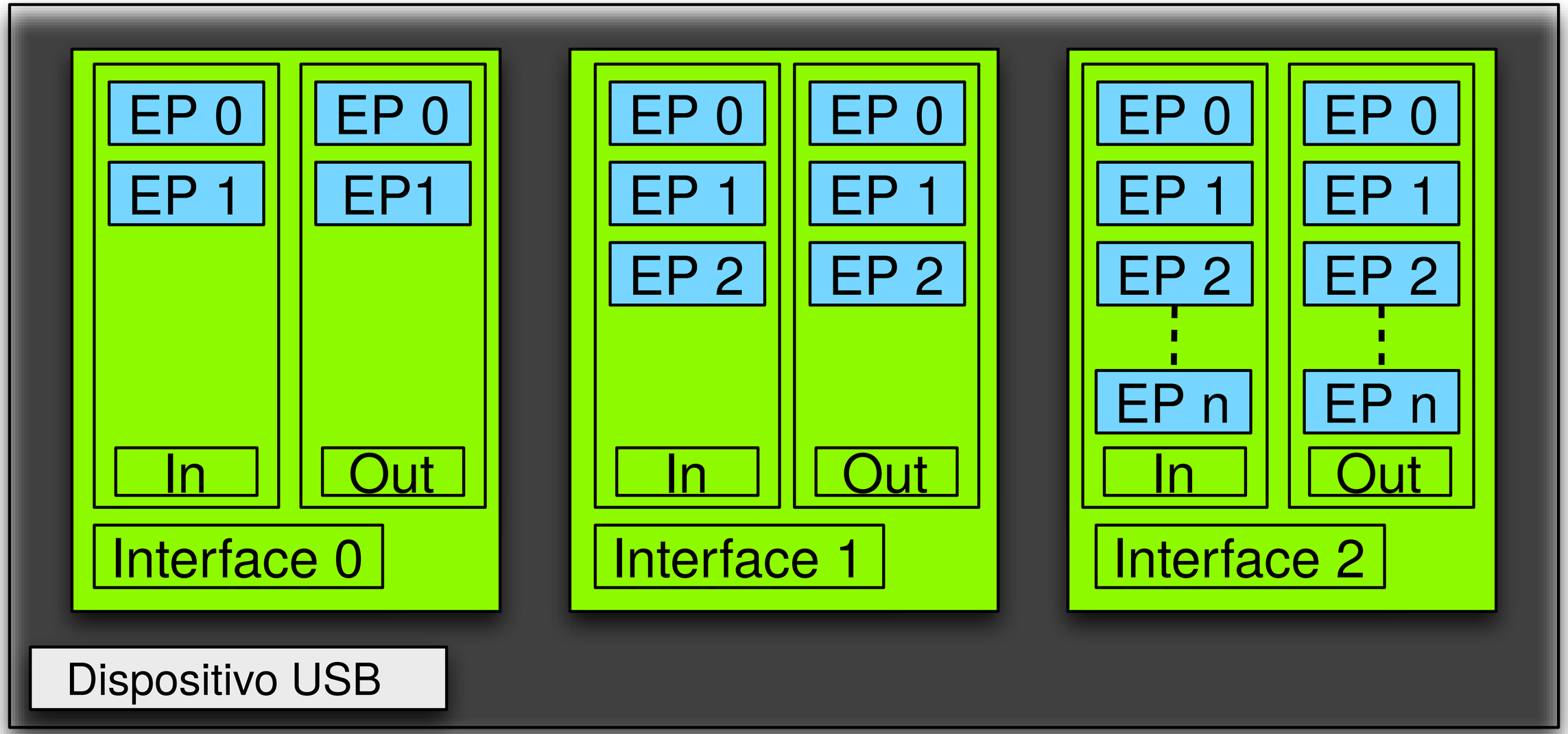
Por que o USB não é mais tão bom?



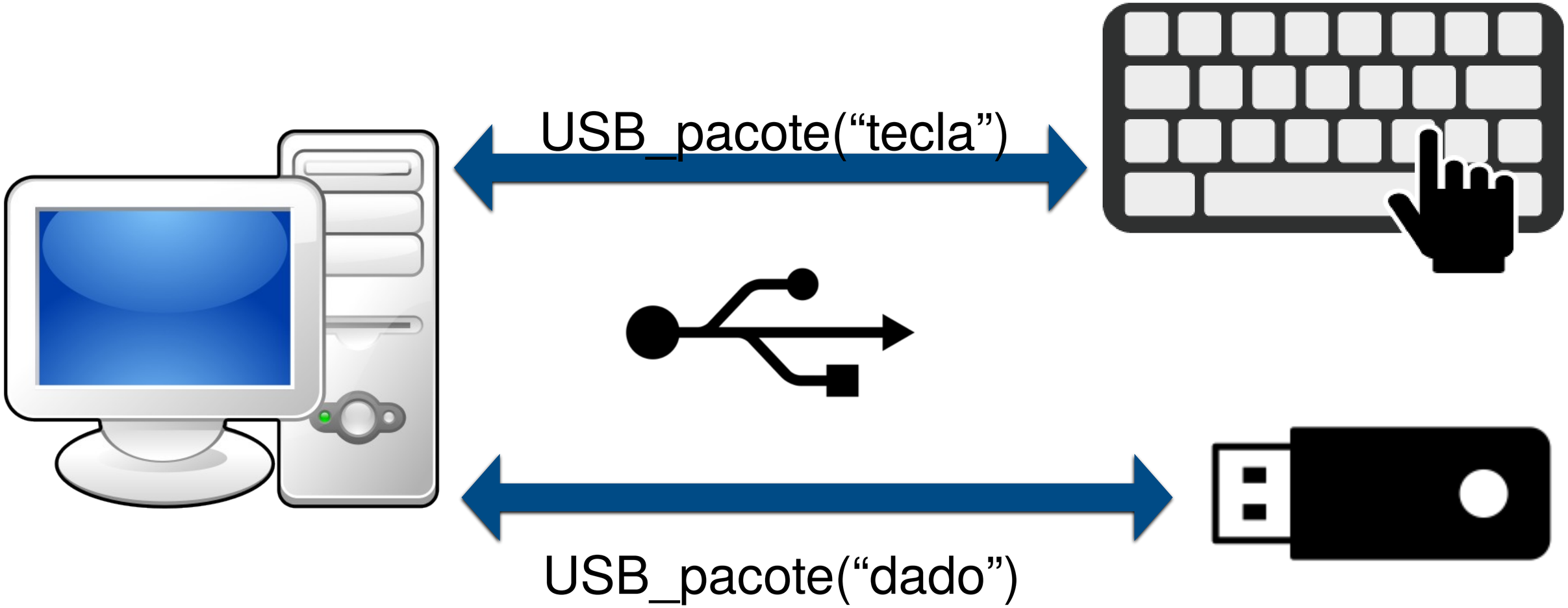
Enumeração USB



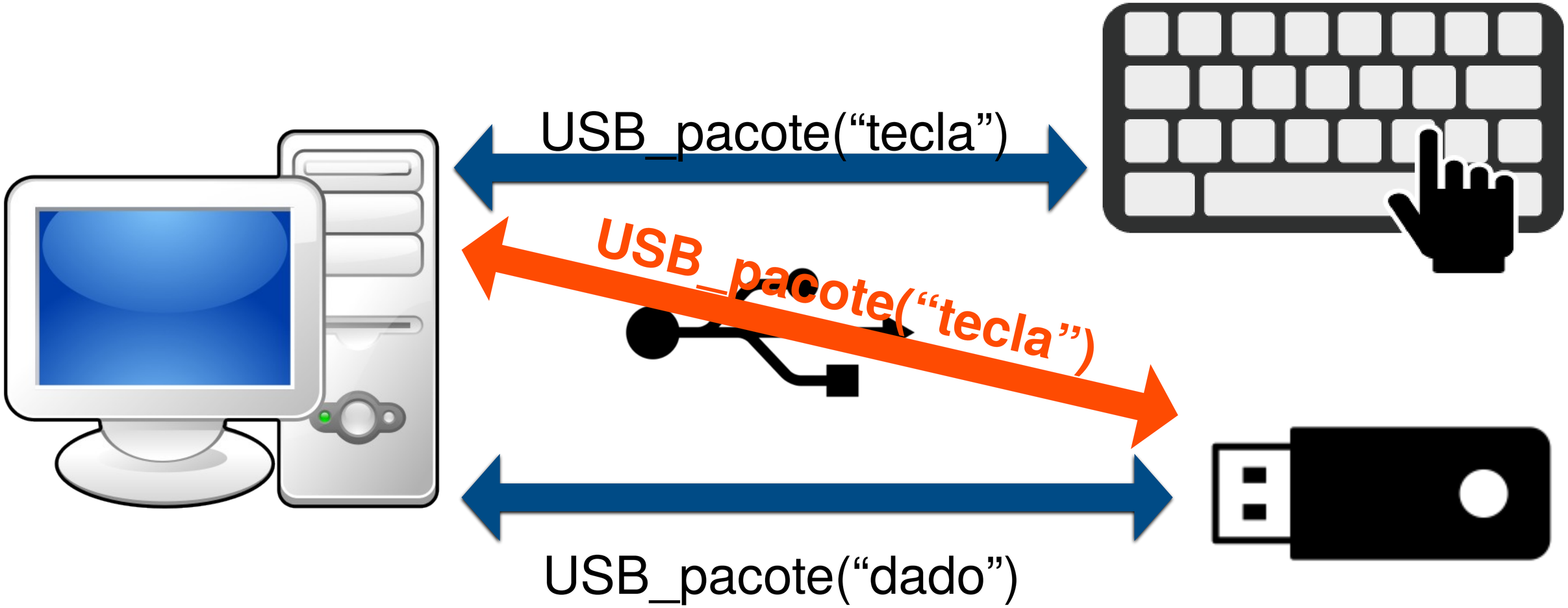
Dispositivo USB



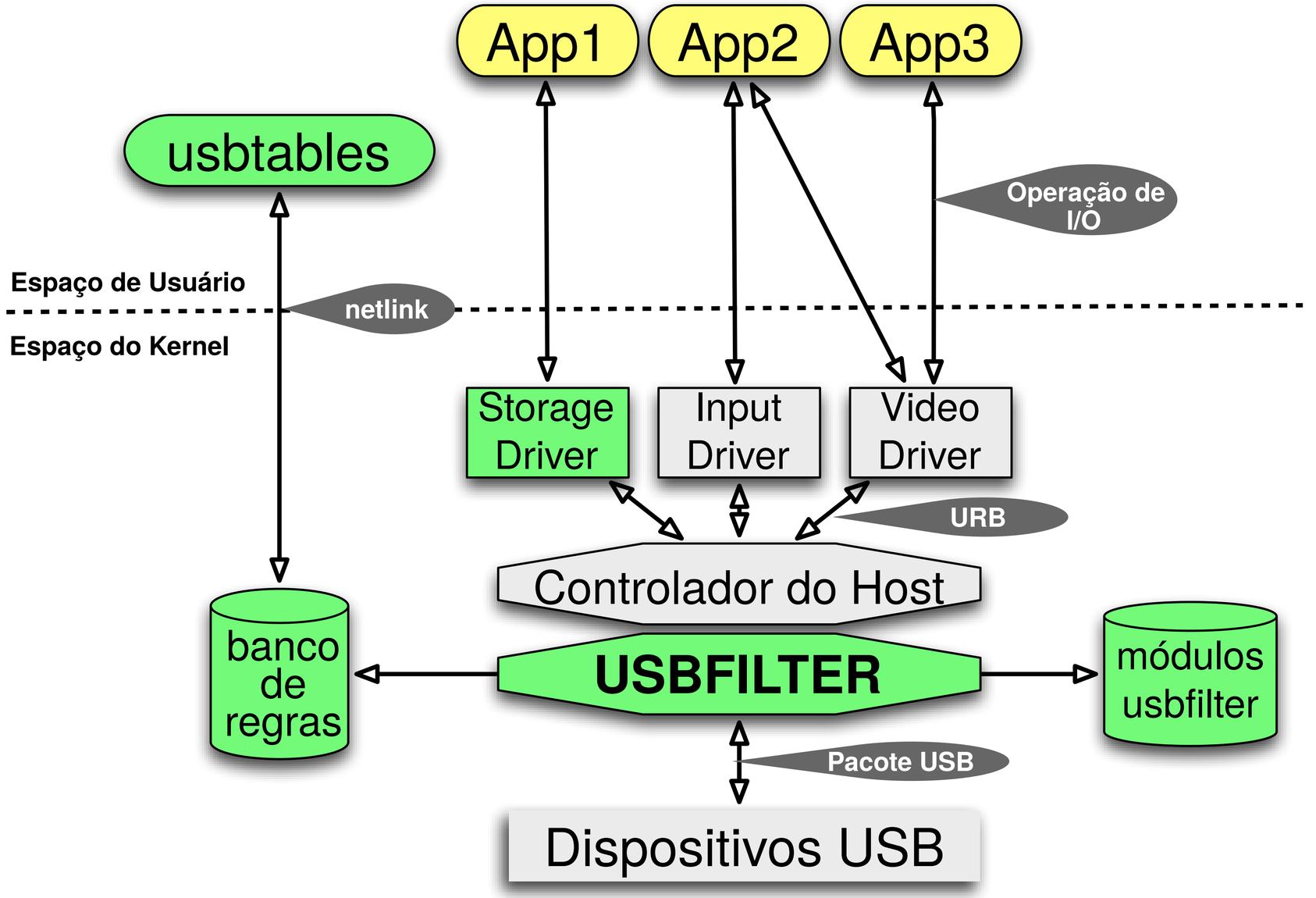
Pacote USB



Pacote USB



USBFILTER


App1
App2
App3
usbtables
Operação de I/O
Espaço de Usuário
netlink
Espaço do Kernel
Storage Driver
Input Driver
Video Driver
URB
Controlador do Host
banco de regras
USBFILTER
módulos usbfilter
Pacote USB
Dispositivos USB

Metas

- Mediação completa
- Inviolável
- Verificável

Monitores de Referência

- Granularidade
- Extensibilidade

Regras de Construção

Processo	pid, ppid, pgid, uid, euid, gid, egid, comm
Dispositivo	bus#, dev#, port#, if#, devpath, manufacturer, product, serial
Pacote	type, direction, endpoint, address
LUM	name

Consistência de Regras

- Conflito Geral

$$conflito\_geral(R_a, R_b) \longleftarrow$$
$$\forall C_i \ni \mathscr{C}:$$
$$(\exists C_i^a \ni R_a \wedge \exists C_i^b \ni R_b \wedge valor(C_i^a) \neq valor(C_i^b)) \vee$$
$$(\exists C_i^a \ni R_a \wedge \nexists C_i^b \ni R_b) \vee$$
$$(\nexists C_i^a \ni R_a \wedge \nexists C_i^b \ni R_b).$$

- Conflito Fraco

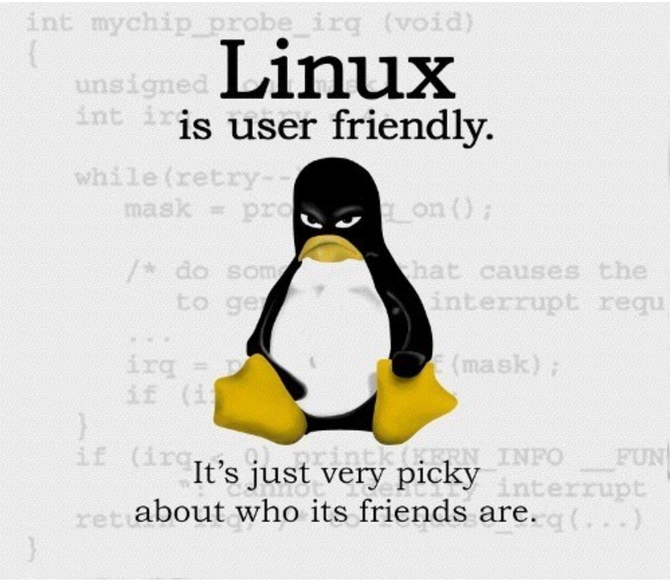
$$conflito\_fraco(R_a, R_b) \longleftarrow$$
$$conflito\_geral\ (R_a, R_b) \wedge\ ação(R_a) = ação(R_b).$$

- Conflito Forte

$$conflito\_forte(R_a, R_b) \longleftarrow$$
$$conflito\_geral\ (R_a, R_b) \wedge\ ação(R_a) \neq ação(R_b).$$

Módulo USBFILTER Linux (LUM)

- Extensão definida pelo usuário para o USBFILTER
  - <linux/usbfilter.h>
- Unidade de construção de regras
  - escrevendo novas regras com LUM
- Olhando dentro do pacote USB
  - comandos SCSI, pacotes IP, pacotes HID, etc.



LUM: detectar escrita pelo SCSI

```
1  int lbsw_filter_urb(struct urb *urb)
2  {
3      char opcode;
4
5      /* HastobeanOUTpacket        */
6      if (usb_pipein(urb->pipe))
7          return 0;
8
9      /* Makesurethepacketislargeenough        */
10     if (urb->transfer_buffer_length<=LUM_SCSI_CMD_IDX)
11         return 0;
12
13     /* Makesurethepacketisnotempty        */
14     if (!urb->transfer_buffer)
15         return 0;
16
17     /* GettheSCSIcmdopcode      */
18     opcode=((  char *)urb->transfer_buffer)[LUM_SCSI_CMD_IDX];
19
20     /* CurrentonlyhandleWRITE_10forKingston        */
21     switch (opcode){
22     case WRITE_10:
23         return 1;
24     default:
25         break;
26     }
27
28     return 0;
29 }
```

Visão Geral

- USBFILTER – 27 arquivos fonte no kernel
  - 4 novos arquivos, 23 arquivos modificados
  - Através do USB, SCSI, Block, e Networking
- USBTABLES
  - Mecanismo Prolog interno
  - 21 regras de construção

USBTABLES -h

```
-d|--debug    Enable debug mode
-c|--config   Path to configuration file(TBD)
-h|--help     Display this help message
-p|--dump     Dump all the rules
-a|--add      Add a new rule
-r|--remove   Remove an existing rule
-s|--sync     Synchronize rules with kernel
-e|--enable   Enable usb filter
-q|--disable  Disable usb filter
-b|--behave   Change the default behavior
-o|--proc     Process table rule
-v|--dev      Device table rule
-k|--pkt      Packet table rule
-l|--lum      LUM table rule
-t|--act      Table rule action
-------------------------------
proc:pid,ppid,pgid,uid,euid,gid,egid,comm
dev:busnum,devnum,portnum,ifnum,devpath,product,
     manufacturer,serial
pkt:types,direction,endpoint,address
lum:name
behavior/action:allow|drop
```

Parando ataques BadUSB

Para teclado/mouse:

```
usbtables -a mymouse -v busnum=1,devnum=4,portnum=2,
      devpath=1.2,product="USBOpticalMouse",
      manufacturer=PixArt-ktypes=1 -t allow
```

```
usbtables -a mykeyboard -v busnum=1,devnum=3,
      portnum=1,devpath=1.1,
      product="DellUSBEntryKeyboard",
      manufacturer=DELL-ktypes=1 -t allow
```

```
usbtables -a noducky -k types=1 -t drop
```

Fixar webcam ao Skype

Para uma webcam Logitech C310:

```
usbtables -a skype -o uid=1001,comm=skype -v
        serial=B4482A20 -t allow
```

```
usbtables -a nowebcam -v serial=B4482A20 -t drop
```

Sem vazamento de dados

Para qualquer dispositivo USB de armazenamento:

```
usbtables -a nodataexfil4
          -l name=block_scsi_write -t drop
```

Headset surdo

Para headsets USB Logitech:

```
usbtables -a logitech-headset -v ifnum=2,product=
        "LogitechUSBHeadset",manufacturer=Logitech -k
        Direction=1 -t drop
```

Carga segura

Para o Nexus 4:

```
usbtables -a n4-charger -v product="Nexus4" -t drop
```

Para qualquer smartphone

```
usbtables -a charger -v busnum=1,portnum=4 -t drop
```

Escalabilidade

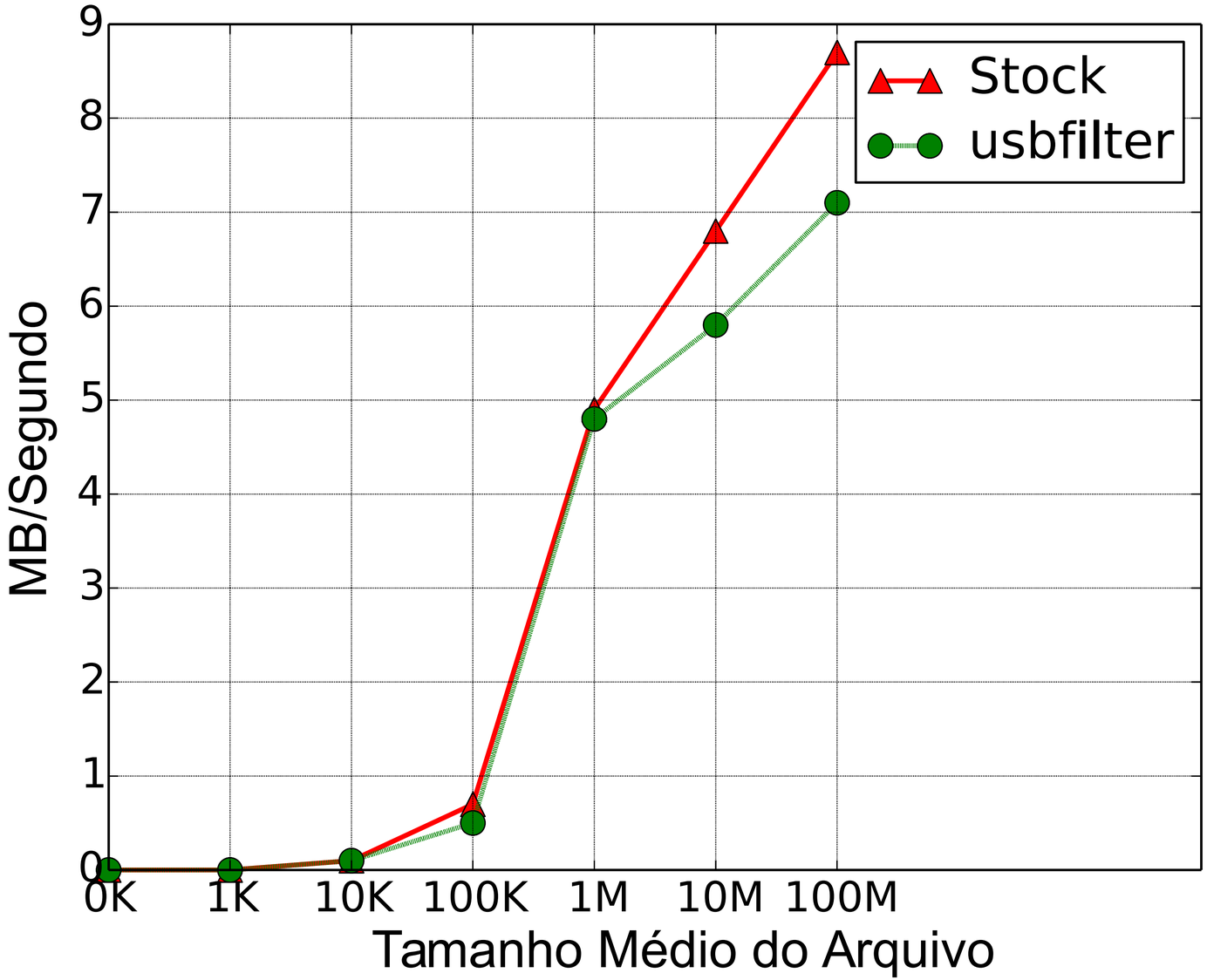
USBTABLES:

Adicionando uma nova regra	Média (ms)
20 Regras	5.9
100 Regras	5.9

USBFILTER:

Filtragem de pacotes	Média (µs)
20 Regras	2.6
100 Regras	9.7

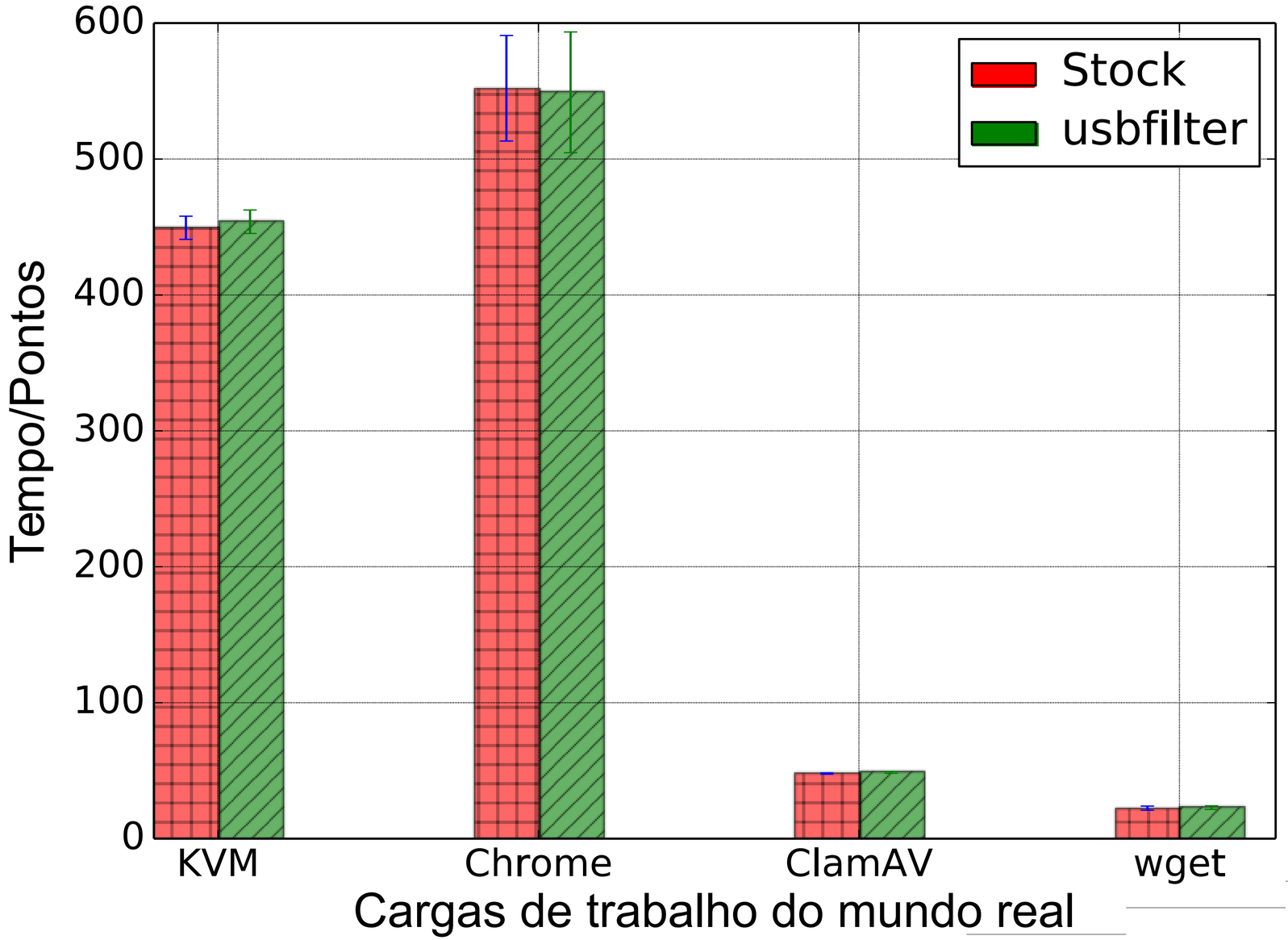
Rendimento


MB/Segundo
0
1
2
3
4
5
6
7
8
9
0K
1K
10K
100K
1M
10M
100M
Tamanho Médio do Arquivo
Stock
usbfilter

Latência

Latência (µs)	1 KB	10 KB	100 KB	1 MB	10 MB	100 MB
Stock	97.6	98.1	99.2	105.5	741.7	5177.7
USBFILTER	97.7	98.2	99.6	106.3	851.5	6088.4
Overhead	0.1%	0.1%	0.4%	0.8%	14.8%	17.6%

Desempenho no mundo real



Limitações e Trabalhos Futuros

- Chamadas de interrupção
- Drivers específicos de vendedor
- Filtrar caminho de resposta
- Criar mais LUMs
- Usabilidade

Download do USBFILTER:
https://github.com/daveti/usbfilter

Informe os bugs em:
root@davejingtian.org