



**REPUBLIQUE DU BENIN**  
**UNIVERSITE D'ABOMEY-CALAVI**  
**(UAC)**  
**ECOLE POLYTECHNIQUE D'ABOMEY CALAVI**  
**(EPAC)**  
**GÉNIE INFORMATIQUE ET TÉLÉCOMS**  
**(4e Année- RII 1)**  
**Matière: Système d'Exploitation Avancé (SEA)**  
  
**RAPPORT DE RECHERCHES SUR LE PWRU**

Réalisé par le groupe 3:

1- HOUENOU Emmanuel

2- TOBOU Charmel

3- YAYA NADJO Sènanmie

Sous la supervision du :

Dr Emery ASSOGBA

Année Académique 2021-2022

## PLAN

- I. INTRODUCTION
- II. VUE D'ENSEMBLE
- III. OBJECTIFS
- IV. QU'EST-CE QUE LE TRAÇAGE DE PAQUETS ?
- V. POURQUOI FAUT-IL TRACER LES PAQUETS?
- VI. LES DIFFÉRENTS TRACEURS DE PAQUETS?
- VII. PRÉSENTATION DE PWRU
- VIII. INSTALLATION DE PWRU
- IX. UTILISATION DE PWRU
- X. COMPARAISON AVEC DES TRACEURS DE PAQUETS
- XI. LIMITES DE PWRU
- XII. CONCLUSION
- XIII. RÉFÉRENCES

-Expliquer les captures. Les observations...

-Faire des tests supplémentaires...

Comparer pwru à d'autres traceurs de paquets...

Limites de pwru...

## I. INTRODUCTION

À bien des égards, le paysage des entreprises numériques a connu une révolution ou deux. Ce qui a commencé comme simple Scripts CGI écrit en Perl a maintenant fleuri dans des déploiements en cluster fonctionnant entièrement automatisés sur Kubernetes et d'autres frameworks d'orchestration. De même, nous découvrons de plus en plus des méthodologies de recherche et de traçage de paquets. Au nombre de celles-ci, figure le packet where are you, plus connu sous le vocable **pwru**. Notre travail consistera à vous en apprendre davantage sur ce dernier et comment on l'utilise dans l'analyse de performances du kernel.

## II. VUE D'ENSEMBLE

L'analyse des performances est essentielle pour tout processus de développement d'une application logicielle.

En plus du débogage, il est nécessaire d'utiliser des pratiques d'instrumentation pour garantir les performances attendues d'une application (ou même d'un noyau). L'instrumentation consiste essentiellement à surveiller et à mesurer les performances, à diagnostiquer les erreurs et à écrire des informations de suivi dans les environnements de développement et de production.

Linux propose des outils d'instrumentation connus sous le nom de traceurs.

### **III. OBJECTIFS**

- Comprendre le système de traçage de paquets
- Être capable de reconnaître un traçage de paquet
- Être en mesure d'utiliser pwrw.

### **IV. QU'EST-CE QUE LES TRACEURS DE PAQUETS ?**

Les traceurs de paquets sont des outils avancés d'analyse et de tests de performances. Ces derniers ne capturent pas seulement que des appels système ou des paquets, mais peuvent généralement tracer n'importe quelle fonction du noyau ou d'une application.

Les traceurs capturent tous les événements qui se produisent dans le noyau (changements de contexte, défauts de pages, interruptions, ....., etc).

### **V. POURQUOI FAUT-IL TRACER LES PAQUETS?**

Il semble que les paquets sont à peu près les éléments de base d'un flux de données réseau. Ce faisant, il s'avère important de procéder au traçage des paquets pour pouvoir:

- Expliquer des pertes inexplicables de données secrètes.
- Diagnostiquer les applications lentes lorsqu'il ne semble y avoir aucune preuve.
- S'assurer que votre ordinateur n'a pas été compromis.

Dans l'ensemble, le traçage des paquets relève de certaines précautions à prendre dans le cadre de la protection des données. Si vous savez comment effectuer un traçage de paquets et une analyse correcte du résultat obtenu, vous pouvez vous éviter d'être accusé à tort d'un piratage ou simplement d'être blâmé en tant que développeur incompetent ou administrateur réseau ou détecter exactement là où il y aurait un problème sur votre réseau ainsi que lors de vos travaux sur votre ordinateur.

La création, le formatage et la lecture des traces de paquets sont parfois nécessaires pour résoudre les problèmes avec IBM, WebSphere, Edge Server. Cependant, l'outil le plus approprié varie en fonction du système d'exploitation.

## VI. LES DIFFÉRENTS TRACEURS DE PAQUETS?

La création, le formatage et la lecture des traces de paquets sont parfois nécessaires pour résoudre les problèmes mais dépendent du système d'exploitation sur lequel on travaille. Ainsi, on distingue:

- ☐ Wireshark est un outil utile et disponible gratuitement qui peut lire des fichiers et capturer des paquets sur presque tous les systèmes d'exploitation.
- ☐ Utilisation de tcpdump sous Linux

Le programme tcpdump possède de nombreuses options et une page de manuel complète.

Les paquets retransmis inattendus capturés dans la trace sont acceptables et n'indiquent pas un problème. Certains scénarios de liaison de canaux entraînent une retransmission et des paquets en double apparaissant sous "-i any". Vous pouvez éliminer les doublons en spécifiant une interface

spécifique, mais il est toujours préférable de capturer autant que possible plutôt que de risquer de manquer des données critiques.

Exemple:

```
tcpdump -s 0 -w filename.pcap
```

If a collocated server is defined:

```
tcpdump -i any -s 0 -w <filename>.pcap
```

- Entrez Ctrl+C pour terminer la trace. Le fichier capture.pcap produit est lisible avec Wireshark
- Pour visualiser le contenu de la capture, utilisez la commande :
  - `tcpdump -r capture.pcap`
- Fournissez le fichier capture.pcap binaire et non formaté à IBM. Si la trace est fournie au format ascii ou texte, une analyse détaillée ne peut pas être effectuée.



## VII. PRÉSENTATION DE PWRU

Pwru est un outil basé sur eBPF pour tracer les paquets réseau dans le noyau Linux avec des capacités de filtrage avancées. Il permet une introspection fine de l'état du noyau pour faciliter le débogage des problèmes de connectivité réseau.

C'est donc un outil qui attache un tas de kprobes à des fonctions ou interfaces spécifiques du noyau , je n'en suis pas si sûr et il montre les endroits dans le chemin de données du noyau qu'un paquet donné traverse

Il a de nombreux formats de sortie mais dans son format de base

`--output-tuple`

(l4 tuples vous montre quelque chose comme ça):

```

0xffff8cf004b00000 [empty] __ip_local_out 2055760703672 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] nf_hook_slow 2055760735769 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] ip_output 2055760750670 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] nf_hook_slow 2055760759326 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] apparmor_ipv4_postroute 2055760771237 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] ip_finish_output 2055760780659 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] __cgroup_bpf_run_filter_skb 2055760792841 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] __ip_finish_output 2055760802904 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] ip_finish_output2 2055760812410 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] dev_queue_xmit 2055760822676 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] __dev_queue_xmit 2055760833310 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] qdisc_pkt_len_init 2055760857829 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] netdev_core_pick_tx 2055760867387 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] sch_direct_xmit 2055760878242 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] validate_xmit_skb_list 2055760903573 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] validate_xmit_skb 2055760915518 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] netif_skb_features 2055760933336 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] skb_network_protocol 2055760953842 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] validate_xmit_vlan 2055760963051 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] validate_xmit_xfrm 2055760972363 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] dev_hard_start_xmit 2055760982137 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] __skb_pad 2055760992970 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [empty] skb_clone_tx_timestamp 2055761007213 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [sshd] __dev_kfree_skb_any 2055761184398 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [sshd] consume_skb 2055761199763 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [sshd] skb_release_head_state 2055761207921 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [sshd] sock_wfree 2055761217061 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [sshd] skb_release_data 2055761225265 10.0.2.15:58990->1.1.1.1:80(tcp)
0xffff8cf004b00000 [sshd] kfree_skbmem 2055761233812 10.0.2.15:58990->1.1.1.1:80(tcp)

```

Figure 1: Fonctions de pwr

Ainsi, vous pouvez voir les fonctions et les tuples des identifiants SKB, il est vraiment très intéressant d'avoir ce niveau de visibilité si facilement, le seul problème est le noyau et les paramètres de configuration qu'il faut utiliser pour l'utiliser, parfois les serveurs de production ne le sont pas. exécutant des noyaux de pointe, ce genre d'outils est donc un peu hors de portée, mais néanmoins, c'est loin d'être génial.

## VIII. INSTALLATION DE PWRU

Pour l'installation de pwru, nous allons suivre les étapes ainsi que la vidéo

## Grandes étapes

*\$go generate*

[illegible]

Figure 2: Génération de go

**Commentaire:** Certains OS exigent une demande permission avant de pouvoir générer go.

```
1 warning generated.
Compiled /home/kaithynd/Documents/pwru/kprobepwru_bpfel.o
Wrote /home/kaithynd/Documents/pwru/kprobepwru_bpfel.go
/home/kaithynd/Documents/pwru/bpf/kprobe_pwru.c:361:21: warning: taking address of packed member 'print_skb_id' of class or structure 'event_t' may
sult in an unaligned pointer value [-Waddress-of-packed-member]
    set_skb_btf(skb, &event->print_skb_id);
                          ^
/home/kaithynd/Documents/pwru/kprobepwru_bpfel.o
1 warning generated.
Compiled /home/kaithynd/Documents/pwru/kprobepwru_bpfel.o
Wrote /home/kaithynd/Documents/pwru/kprobepwru_bpfel.go
/home/kaithynd/Documents/pwru/bpf/kprobe_pwru.c:361:21: warning: taking address of packed member 'print_skb_id' of class or structure 'event_t' may
sult in an unaligned pointer value [-Waddress-of-packed-member]
    set_skb_btf(skb, &event->print_skb_id);
                          ^
/home/kaithynd/Documents/pwru/kprobepwruwithoutoutputskb_bpfel.o
1 warning generated.
Compiled /home/kaithynd/Documents/pwru/kprobepwruwithoutoutputskb_bpfel.o
Wrote /home/kaithynd/Documents/pwru/kprobepwruwithoutoutputskb_bpfel.go
/home/kaithynd/Documents/pwru/bpf/kprobe_pwru.c:361:21: warning: taking address of packed member 'print_skb_id' of class or structure 'event_t' may
sult in an unaligned pointer value [-Waddress-of-packed-member]
    set_skb_btf(skb, &event->print_skb_id);
                          ^
/home/kaithynd/Documents/pwru/kprobepwruwithoutoutputskb_bpfel.o
1 warning generated.
Compiled /home/kaithynd/Documents/pwru/kprobepwruwithoutoutputskb_bpfel.o
Wrote /home/kaithynd/Documents/pwru/kprobepwruwithoutoutputskb_bpfel.go
```

**Figure 3:** Génération de go avec sudo

### *\$go build*

On rencontre le même problème que précédemment, avec un problème de permission.





Commentaire: On remarque que pour pouvoir mener à bien cette installation, nous devons avoir la version 1.17 de go.

NB: Installation de go 1.17.5(mise à jour vers la version 1.17)

Nous avons déjà eu à installer go dans le but de l'accomplissement de la tâche liée à ndt7-monitor. Il nous faudra juste vers une mise à jour vers la nouvelle version .

### ★ \$go generate

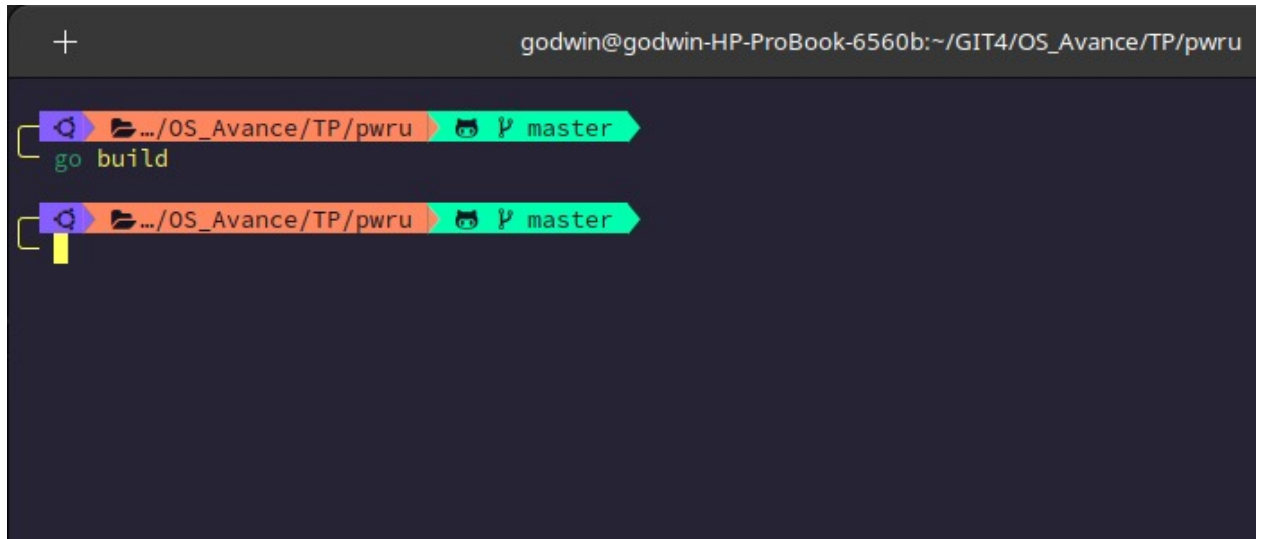
```

godwin@godwin-HP-ProBook-6560b:~/GIT4/OS_Avance/TP/pwru
└─$ go generate
/home/godwin/GIT4/OS_Avance/TP/pwru/bpf/kprobe_pwru.c:362:21: warning: taking address of packed member 'print_skb_id' of
class or structure 'event_t' may result in an unaligned pointer value [-Waddress-of-packed-member]
    set_skb_btf(skb, &event->print_skb_id);
                          ^
1 warning generated.
Compiled /home/godwin/GIT4/OS_Avance/TP/pwru/kprobepwru_bpfel.o
Wrote /home/godwin/GIT4/OS_Avance/TP/pwru/kprobepwru_bpfel.go
/home/godwin/GIT4/OS_Avance/TP/pwru/bpf/kprobe_pwru.c:362:21: warning: taking address of packed member 'print_skb_id' of
class or structure 'event_t' may result in an unaligned pointer value [-Waddress-of-packed-member]
    set_skb_btf(skb, &event->print_skb_id);
                          ^
1 warning generated.
Compiled /home/godwin/GIT4/OS_Avance/TP/pwru/kprobepwru_bpfeb.o
Wrote /home/godwin/GIT4/OS_Avance/TP/pwru/kprobepwru_bpfeb.go
/home/godwin/GIT4/OS_Avance/TP/pwru/bpf/kprobe_pwru.c:362:21: warning: taking address of packed member 'print_skb_id' of
class or structure 'event_t' may result in an unaligned pointer value [-Waddress-of-packed-member]
    set_skb_btf(skb, &event->print_skb_id);
                          ^
1 warning generated.
Compiled /home/godwin/GIT4/OS_Avance/TP/pwru/kprobepwruwithoutoutputskb_bpfel.o
Wrote /home/godwin/GIT4/OS_Avance/TP/pwru/kprobepwruwithoutoutputskb_bpfel.go
/home/godwin/GIT4/OS_Avance/TP/pwru/bpf/kprobe_pwru.c:362:21: warning: taking address of packed member 'print_skb_id' of
class or structure 'event_t' may result in an unaligned pointer value [-Waddress-of-packed-member]
    set_skb_btf(skb, &event->print_skb_id);
                          ^
1 warning generated.
Compiled /home/godwin/GIT4/OS_Avance/TP/pwru/kprobepwruwithoutoutputskb_bpfeb.o
Wrote /home/godwin/GIT4/OS_Avance/TP/pwru/kprobepwruwithoutoutputskb_bpfeb.go

```

Figure 6: Résultat obtenu avec succès du go generate

★ *\$go build*

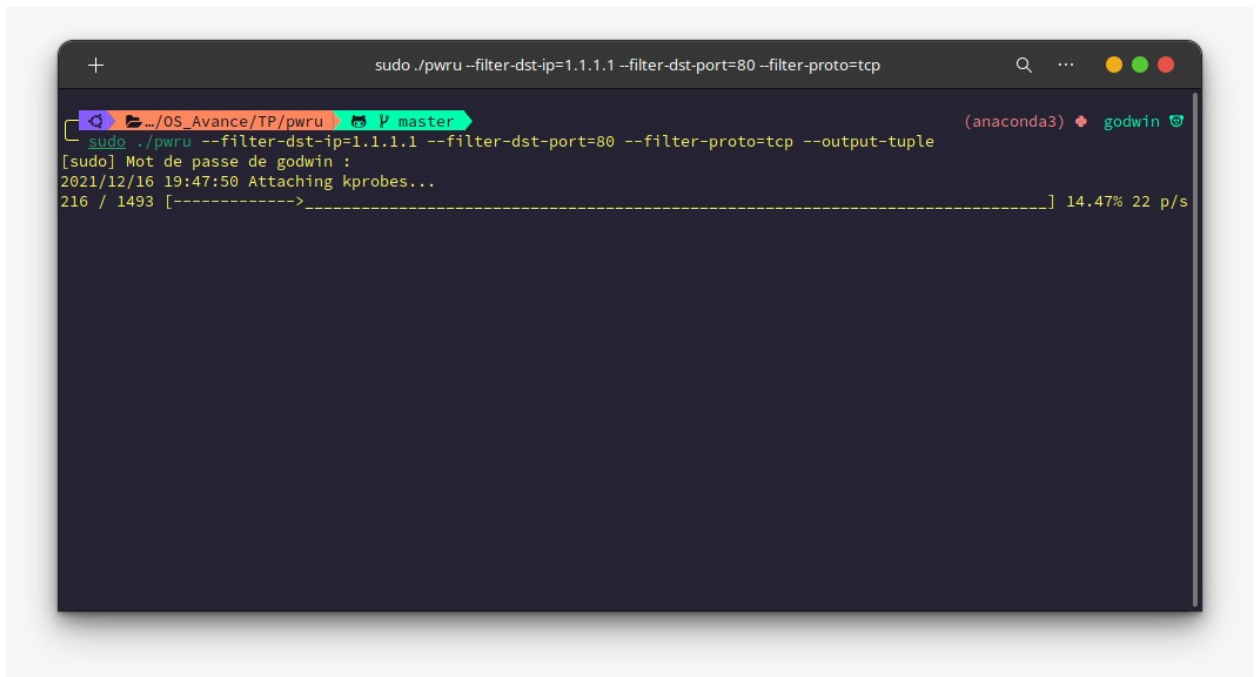


```
godwin@godwin-HP-ProBook-6560b:~/GIT4/OS_Avance/TP/pwru
[ ] go build
```

The image shows a terminal window with a dark background. At the top, the prompt is 'godwin@godwin-HP-ProBook-6560b:~/GIT4/OS\_Avance/TP/pwru'. Below the prompt, there are two lines of text, each preceded by a yellow bracket '['. The first line is 'go build' and the second line is also 'go build'. Each line is preceded by a small icon of a gear and a folder, and followed by a green bar containing a cat icon and the word 'master'.

Figure 7: Résultat obtenu avec succès du go build

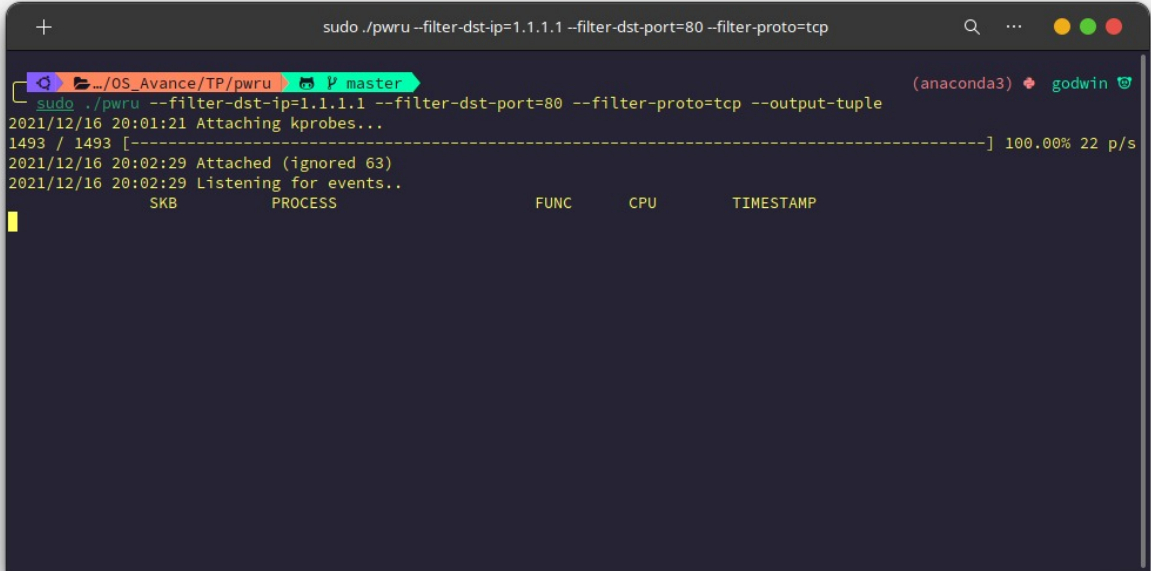
## ★ *\$filtrage de packet*



```
sudo ./pwru --filter-dst-ip=1.1.1.1 --filter-dst-port=80 --filter-proto=tcp
[anaconda3] godwin
[sudo] Mot de passe de godwin :
2021/12/16 19:47:50 Attaching kprobes...
216 / 1493 [----->] 14.47% 22 p/s
```

Figure 8: Résultat obtenu avec la commande de filtrage





```
sudo ./pwru --filter-dst-ip=1.1.1.1 --filter-dst-port=80 --filter-proto=tcp
sudo ./pwru --filter-dst-ip=1.1.1.1 --filter-dst-port=80 --filter-proto=tcp --output-tuple
2021/12/16 20:01:21 Attaching kprobes...
1493 / 1493 [-----] 100.00% 22 p/s
2021/12/16 20:02:29 Attached (ignored 63)
2021/12/16 20:02:29 Listening for events..
      SKB      PROCESS      FUNC      CPU      TIMESTAMP
```

Figure 9: Résultat avec le début de filtrage

```

+ sudo ./pwru --filter-dst-ip=1.1.1.1 --filter-dst-port=80 --filter-proto=tcp
[anaconda3] godwin
[ ] sudo ./pwru --filter-dst-ip=1.1.1.1 --filter-dst-port=80 --filter-proto=tcp --output-tuple
[sudo] Mot de passe de godwin :
2021/12/16 19:47:50 Attaching kprobes...
1493 / 1493 [-----] 100.00% 22 p/s
2021/12/16 19:49:00 Attached (ignored 63)
2021/12/16 19:49:00 Listening for events..
      SKB      PROCESS      FUNC      CPU      TIMESTAMP
0xffff96e04e3154e0 [curl] ip_local_out      2      2575551309866 192.168.43.131:52866->1.1.1.1:80(
tcp)
0xffff96e04e3154e0 [curl] __ip_local_out  2      2575551345979 192.168.43.131:52866->1.1.1.1:80(
tcp)
0xffff96e04e3154e0 [curl] nf_hook_slow   2      2575551358139 192.168.43.131:52866->1.1.1.1:80(
tcp)
0xffff96e04e3154e0 [curl] ip_output       2      2575551401059 192.168.43.131:52866->1.1.1.1:80(
tcp)
0xffff96e04e3154e0 [curl] nf_hook_slow   2      2575551410342 192.168.43.131:52866->1.1.1.1:80(
tcp)
0xffff96e04e3154e0 [curl] apparmor_ipv4_postroute 2      2575551421228 192.168.43.131:52866->1.1.1.1:80(
tcp)
0xffff96e04e3154e0 [curl] ip_finish_output 2      2575551444986 192.168.43.131:52866->1.1.1.1:80(
tcp)
0xffff96e04e3154e0 [curl] __cgroup_bpf_run_filter_skb 2      2575551455322 192.168.43.131:52866->1.1.1.1:
80(tcp)
0xffff96e04e3154e0 [curl] __ip_finish_output 2      2575551463957 192.168.43.131:52866->1.1.1.1:80(
tcp)

```

Figure 10: Résultat obtenu avec le filtrage

```

+ curl 1.1.1.1
[anaconda3] godwin
[ ] sudo iptables -t filter -I OUTPUT 1 -m tcp --proto tcp --dst 1.1.1.1/32 -j DROP
[anaconda3] godwin
[ ] curl 1.1.1.1

```

Figure 11: Résultat obtenu avec 1.1.1.1

## IX. CONCLUSION

Les systèmes d'exploitation de Linux ou Mac OS X se basent sur Unix et possèdent des outils de commande sur lesquelles l'utilisateur peut conduire le système avec l'aide d'un terminal via une saisie. Il existe de nouvelles versions qui permettent de rassembler les outils de diagnostics importants avec une interface utilisateur design. C'est ainsi qu'on a le pwr pour le traçage des paquets des programmes de service réseau. La façon la plus simple pour parvenir au service réseau est d'utiliser la fonction Spotlight qui, après la saisie « programmes de service réseau »

## X. RÉFÉRENCES

<https://bearstech.com/societe/blog/decouvrez-ce-que-le-kernel-a-dans-le-ventre-avec-eBPF/>

<https://www.tigera.io/learn/guides/ebpf/>

<https://brendangregg.com/blog/2019-01-01/learn-ebpf-tracing.html>

<https://newrelic.com/fr/blog/best-practices/what-is-ebpf>

<https://medium.com/smileinnovation/les-traceurs-sous-linux-1-2-349f4e221e9d>

