eBPF bumblebee simple example

<https://bumblebee.io/EN> -> Bumblebee with simple example

C, Network, Hashmap, print -> probe.c

* sudo -s
* apt install curl
* apt install docker.io
* curl -sL https://run.solo.io/bee/install | sh -> install Command Line Interface
* export PATH=$HOME/.bumblebee/bin:$PATH .> add bee CLI to path
* bee init
* C, Network (o filesystem), HashMap (o RingBuffer), print (o counter o gauge)

Programmi eBPF iniettati in kernel e usati per cambiare dinamicamente le capacità dello stack del network, del filesystem,…

Caso in cui si raccolgono informazioni su ciò che accade sul network.

* probe.c

si avrà un file .c con mappe create, connessione con tcp, eBPF programma base che conta il numero di volte che un processo accede al network.

Per compilare eseguire

* bee build probe.c probe:v1 // probe:v1 è il nome

si impacchetta il programma eBPF come un’immagine OCI.

Questa immagine si può condividere con altri utenti, deployare fuori dall’ambiente,…

Per vedere le immagini OCI già create che contengono un programma eBPF

* bee list

Immagine che contiene testo

Descrizione generata automaticamente

per eseguire il programma eBPF

* bee run probe:v1

compare a terminale il programma in user-space.

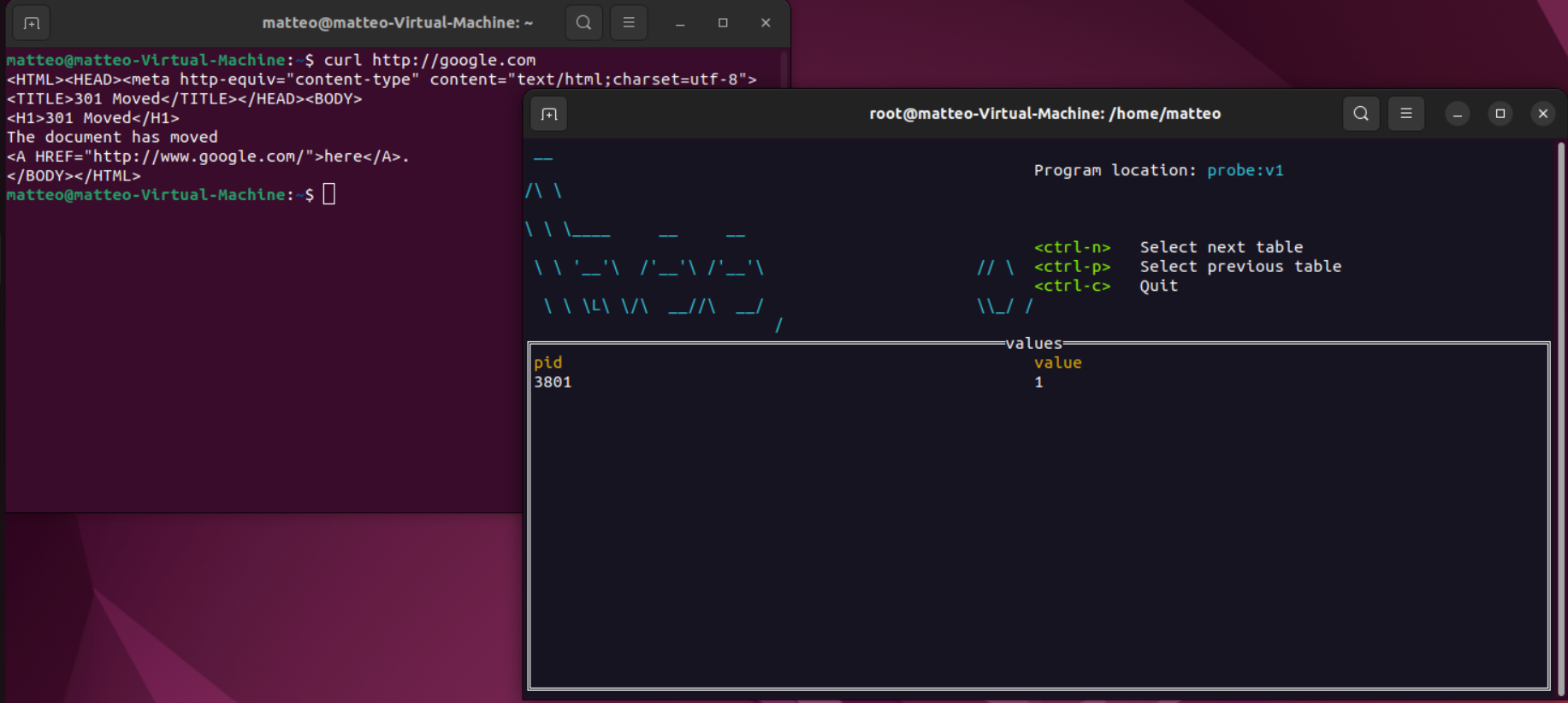
Il comando di print serve per mostrare tutti i dati raccolti nelle mappe.

Se da un’altra finestra si digita

* curl *indirizzo\_http*

si osserva il funzionamento del programma:

* collezione l’ID del processo (pid)
* conta il numero di volte che quel processo si connette al network (value)



Per ripetere l’esecuzione nella macchina virtuale ricordarsi di

* curl -sL https://run.solo.io/bee/install | sh
* export PATH=$HOME/.bumblebee/bin:$PATH