FIREWALL RULE CONVERTER

Damiano Barone
Lorenzo Diana

Abbiamo implementato un algoritmo scritto in python che prende delle regole di un router (file iptable.py) e li converte in una serie di funzioni che si richiamo a cascata, imitando il comportamento di uno switch.
Nel file Iptable.py abbiamo creato una sintassi ben precisa per stabilire come scrivere le regole. I campi disponibili sono protocol, srcport, srcip, destport, destip. Si devono specificare tutti i campi. Tra una parola e l'altra deve esserci uno spazio. Per gestire il campo protocollo si deve scrivere 0-0 se tcp, altrimenti 1-1 se udp. Per mettere un range di valori utilizzare il carattere '-' , ad esempio se una regola accetta protocolli sia tcp che udp si deve inserire 0-1. Inoltre va messa l'azione della regola prima della regola stessa, si posso raggruppare le regole che hanno la stessa azione. L'azione puo' essere accept o reject ecco un esempio:

```
ACCEPT
protocol = 1-1 srcport = 500-600  srcip = 192.169.1.1-192.170.1.1 destport = 5001-6000 destip = 192.168.1.1-192.168.1.1
```

Inoltre si deve pure inserire il campo default che corrisponde all'azione sul pacchetto che non ha fatto match con nessuna regola. Per vedere un esempio concreto basta vedere il file iptable.py del progetto.
Nel caso una regola abbia un range che si sovrappone ad un range di un'altra regola il programma segnala un conflitto e termina senza generare il file di output (control.c). Si noti che se 2 regole hanno lo stesso campo min e max non e' conflitto. Il file packet.h contiene la struttura di tutti i campi.
Il file test.c contiene un esempio di utilizzo della funzione check(), essa restituirà se il pacchetto e' stato accetto a rifiutato del firewall.
L'algoritmo che implementa la funzione e' nel file Firewall_rule_converter.py nella prima fase viene controllato la sintassi del file iptable e vengono inserite in una matrice composta da 11 colonne (2 colonne per ogni valore min-max dei campi, più l'ultima colonna che definisce l'azione da fare). Successivamente va ordinata la matrice, cosi' viene piu' facile controllare se ci sono conflitti. L'ultimo passo e' quello di creare il file control.c con tutte le funzioni. Successivamente si può eseguire il test.c per vedere se funziona correttamente.

Per utilizzare l'algoritmo basta eseguire lo script in python (versione 3.5) e dare il comando make tutto da linea di comando

```
[damianos-MacBook-Pro:Firewall_converter Damiano$ python firewall_rule_converter.py
control.c creato corretemente.

[damianos-MacBook-Pro:Firewall_converter Damiano$ make
cc -c -O2 -Wall -Werror control.c
cc -o test test.c control.o -O2 -Wall -Werror
[damianos-MacBook-Pro:Firewall_converter Damiano$ ./test
0
```