***Telcolab - FIREWALL***

di Riccardo Frigerio – 781315

L’idea dell’applicazione è quella di acquisire una richiesta da parte dell’utente tramite settaggio di tre variabili statiche intere costanti, contenute nel sorgente Java MonitorHandler.java. Tali costanti indicano la banda di traffico non oltrepassabili, i cicli consecutivi di non rispetto di tale limite, il tempo per cui il ban deve rimanere attivo.

L’applicazione otterrà la topologia della rete volta per volta, ogni primo ping da parte di un host, ONOS include quest’ultimo nella topologia complessiva in modo da fornire all’applicazione una situazione sempre aggiornata. L’applicazione costruirà quindi le regole per l’instradamento di tutti i pacchetti nella rete e, successivamente, entrerà in un ciclo infinito in cui continuerà a monitorare il traffico generato dagli host ed eventualmente li bloccherà se non rispettano le regole imposte.

Il blocco sarà rimosso dopo BAN\_TIME secondi. In questo intervallo l’applicazione continua il monitoraggio per gli altri host.

L’applicazione è terminabile tramite CLI di ONOS.

Di seguito la spiegazione di ogni file sorgente.

Firewall.java

Definisce lo scheletro dell’applicazione, acquisendo da ONOS tutti i servizi necessari al corretto funzionamento e implementando i due tipi di Runnable su cui l’applicazione si fonda:

* MonitorRunnable: ci sarà una sola istanza di questa classe e sarà responsabile del monitoraggio tramite un oggetto di tipo MonitorHandler. Tale istanza verrà terminata alla chiusura dell’operazione
* FirewallRunnable: ci saranno tante istanze di questa classe quanti saranno i blocchi temporanei. Ognuna di queste sarà terminata dopo 15 secondi.

Questa classa, che non viene mai istanziata, definisce anche tutto il meccanismo di comunicazione con i servizi di ONOS, per il corretto instradamento dei pacchetti e il blocco degli host che generano DoS. A questo scopo, occorre fornire una definizione personalizzata che implementi concretamente la classe PacketProcessor.

Tale classe concreta di chiama CustomPacketProcessor e effettua l’override del metodo process(), responsabile del processo di ogni pacchetto che viene ricevuto dai servizi di ONOS.

Successivamente, la classe provvede alla creazione di regole (tramite intents), se non già esistenti, altrimenti al semplice inoltro dei pacchetti.

La logica di blocco e sblocco dovuti al non rispetto dei limiti imposti dal Firewall è implementata dai metodi drop() e undrop().

Gli ulteriori due metodi da implementare obbligatoriamente sono activate() e deactivate(), i quali definiscono le operazioni da effettuare alla ricezione dell’evento di attivazione (avviare il thread del monitoraggio e rimanere in ascolto di pacchetti in ingresso) e disattivazione dell’applicazione stessa (terminazione di tutti i thread e disiscrizione dai servizi di ONOS).

MonitorHandler.java

Definisce l’implementazione dell’oggetto responsabile del monitoraggio, il cui comportamento è influenzato dal settaggio, da parte dell’utente, delle costanti spiegate nell’introduzione di questa piccola documentazione. Tale classe, istanziata una sola volta dalla classe Firewall, monitora il traffico prodotto da ogni host e invia un segnale ogni volta che un nuovo host supera i limiti. Questo porterà la classe Firewall a creare un thread dedicato per la gestione di tale problema.

La documentazione su tutte le classi e le API di ONOS è reperibile all’indirizzo <http://api.onosproject.org/1.2.1/org/onosproject> e relative URL derivate.