HTTP Monitor

progetto per l'esame Gestione di Rete
A.A. 2018/2019 di

Simone Ricci, 545857

UNIVERSITÀ DI PISA

1. Descrizione

HTTP Monitor è un semplice tool che monitora il traffico HTTP (e HTTPS), generato dall'host su cui il programma viene eseguito, producendo un report su terminale che mostra il numero di bytes e di pacchetti (inviati, ricevuti e totali) e il numero di richieste e di risposte HTTP raggruppate per tipo.

N.B.: la parte relativa al tipo di richiesta effettuata e al tipo di risposta ricevuta viene aggiornata solamente in caso di pacchetti HTTP. Per i pacchetti HTTPS il tool si limita semplicemente ad incrementare il numero di bytes e di pacchetti visti.

2. Funzionamento

Per avviare il tool digitare *[sudo] python3 HTTPMonitor.py*. Il programma riceve in input da command line un solo argomento obbligatorio che può essere:

- il tempo per cui il tool deve sniffare i pacchetti, preceduto da "-t". Per default il tempo è considerato essere in secondi, ma utilizzando l'opzione "-M" o "-H" si può dare in input al programma un tempo espresso in minuti o in ore;
- un file pcap, generato precedentemente mediante tools esterni (i.e. wireshark), da cui leggere i pacchetti ed effettuare l'analisi. Tale parametro è preceduto dall'opzione "-f".

Examples:
1) sudo python3 HTTPMonitor.py -t 20
2) sudo python3 HTTPMonitor.py -f ./capture.pcap
Nel primo caso vengono sniffati pacchetti per 20 secondi, al termine dei quali viene stampato il report. Nel secondo caso, invece, il programma legge il file capture.pcap ed estrae uno ad uno tutti i pacchetti memorizzati in esso. Dopo aver estratto e analizzato l'ultimo pacchetto del file viene stampato il report.

2.1 Altre opzioni

Il programma può essere avviato anche con altre opzioni, oltre a quelle appena viste:

- "*-i interface*" specifica l'interfaccia su cui sniffare i pacchetti (default *eth0);*

- *“-r”* abilita la risoluzione degli indirizzi ip al momento della stampa del report
- “*-v*” abilita la modalità verbose

3. Implementazione

HTTP Monitor è interamente scritto in python (versione 3.7). Utilizza la libreria *scapy* permette la manipolazione e l’analisi di pacchetti nei primi 3 livelli (livello ethernet, livello rete e livello trasporto). Per poter analizzare e manipolare anche il livello applicazione, in particolare il livello HTTP, è necessaria un’estensione della libreria, denominata *scapy-http*.
L’aggiornamento delle statistiche viene effettuato mediante una gerarchia di classi (v. file *HTTPStats.py*).
La classe HTTPStats mantiene statistiche relative al traffico HTTP (e HTTPS) rilevato tra l’host locale e un certo host remoto, separando i pacchetti tra inviati e ricevuti. Nel momento in cui viene aggiunto un pacchetto vengono incrementati il numero di bytes e pacchetti totali scambiati tra i due host, successivamente si incrementano i valori relativi ai pacchetti inviati o ricevuti. Per l’aggiornamento delle statistiche riguardanti il numero di metodi HTTP utilizzati dall’host locale e il tipo di risposte ricevute la classe mantiene due oggetti di tipo *RcvdPacketStats* e *SentPacketStats* che mantengono un dizionario ciascuno dove memorizzano, rispettivamente, le risposte ricevute e il tipo di richieste effettuate. Le chiavi per accedere al dizionario sono stringhe, ovvero i nomi dei metodi o delle risposte, e i valori associate sono il numero di pacchetti rilevati corrispondenti a quella determinata chiave. Inoltre è necessario che il pacchetto abbia un livello HTTP che contenga informazioni circa la richiesta e la risposta. Se ciò non avviene (o perché si tratta di un pacchetto HTTPS, o perché tali informazioni erano contenute in un pacchetto precedentemente rilevato o perché si tratta di un pacchetto che non trasporta informazioni) allora vengono semplicemente incrementati il numero di pacchetti e di bytes scambiati.
All’avvio lo script analizza gli argomenti passati tramite command line:

- nel caso in cui si desideri effettuare un’analisi del traffico che avviene al momento (e quindi se è stata passata l’opzione *-t),* viene creato un thread che ha lo scopo di “tirare su” dalla scheda di rete tutti i pacchetti HTTP (o HTTPS) destinati o inviati all’host locale, utilizzando un filtro *BPF,* e aggiungerli ad un oggetto di tipo

ClientStats che agisce come descritto sopra. Il main-thread effettua una *sleep* per il tempo desiderato, dopo di ché ferma il thread sniffer e stampa a terminale le statistiche ricavate;

- se invece si desidera effettuare un analisi del traffico “offline” allora il main thread si limita a leggere il file *.pcap* e ad inserire i pacchetti nella struttura dati, per poi effettuare la stampa una volta terminata la lettura.

4. Test e dipendenze

Il file *HTTPMonitor_test.py* contiene uno script che implementa un semplice test per il tool *HTTP Monitor.* Tale script si limita a generare del traffico HTTP, mediante richieste casuali, verso gli host scritti nel file *test_list.txt.*
Per testare il programma (avviando sia il tool che il test) lanciare il comando *[sudo] ./test.sh interface*.

Per poter eseguire *HTTP Monitor* è necessario che sia istallata la libreria scapy (*pip3 install scapy*) e la sua estensione per HTTP, scapy-http (*pip3 install scapy-http*).
Inoltre test.sh utilizza il comando da terminale http (*apt-get install httpie*).