Gianluca Medici 275788

Relazione conclusiva del progetto GeoPacketVisualizer per SGR 2008/2009

1: Introduzione

Il progetto GeoPacketVisualizer consiste in un piccolo programma scritto in C per l'ambiente GNU/Linux che molto semplicemente analizza e visualizza su di una mappa del mondo la provenienza e la destinazione di datagrammi IP; questi datagrammi gli vengono forniti attraverso un file pcap contenente una sequenza di pacchetti sniffati su una qualsiasi interfaccia.
Il programma è stato scritto interamente in ambiente GNU/Linux, utilizzando Eclipse v 3.4.2 come IDE per la scrittura del codice e per il debugging.
Gli altri strumenti utilizzati per la realizzazione del programma sono:

- libpcap v 1.0 per il poter accedere ai dati del savefile
- GeoIP v 1.4.6 (con l'aggiunta del DB GeoLite City) che permette di geo referenziare gli IP trovati
- UtHash v 1.6 struttura dati hash utilizzata nell'analisi e nella presentazione dei risultati
- Google Visualization Chart e Table per la visualizzazione su browser web dei dati raccolti.

1.1:Struttura dei file

Il progetto è totalmente contenuto nella cartella GeoPacketVisualizer comprende i seguenti file:

Debug/	
DEBUG	istruzioni per la compilazione e il debugging
makefile	
objects.mk	
sources.mk	
subdir.mk	
analyzer.c	funzioni per l'analisi dei pacchetti “catturati” da libpcap
analyzer.h	
GeoVis.c	main e funzioni di utilità
GeoVis.h	
INSTALL	istruzioni per la compilazione e l'installazione
makefile	
myhash.c	funzioni per la manipolazione della tabella hash
myhash.h	
objects.mk	
README	
resultEXAMPLE.htm	pagina di esempio dei risultati
sources.mk	
subdir.mk	
sysmacro.h	macro di utilità
uthash.h	implementazione della hashtable interamente mediante macro
webStuff.c	funzioni per la visualizzazione dei risultati su browser
webStuff.h	

2: Strutture dati

Solo una struttura dati è stata utilizzata in questo progetto, si tratta della tabella hash che serve a mantenere le informazioni ricavate dal database GeoIP (in realtà da GeoLite City), per questo motivo sono state create delle funzioni di inserimento, ricerca, cancellazione e ordinamento specifiche per i dati che questa hash doveva contenere. La particolarità di UtHash, oltre alla sua estrema leggerezza e semplicità, è quella di essere completamente realizzata tramite macro.

3: Struttura di GeoPacketVisualizer

GeoPacketVisualizer è estremamente semplice nella sua struttura, le operazioni che deve svolgere sono (in ordine) : analisi del savefile pcap, ricerca per ogni datagramma IP trovato degli IP addresses nel database GeoLite City, inserimento in una struttura dati veloce per poter ordinare e contare i dati, iterazione sull'intera struttura dati e invio di una pagina html su socket al browser per la visualizzazione.
Il programma in se è single-threaded, sia per la sua semplicità di realizzazione e di debugging, sia per la sequenzialità delle operazioni che deve svolgere; il main è contenuto nel file GeoVis.c, per prima cosa viene aperto il file pcap passato con l'opzione -p, si compila e si setta un filtro pcap che permette l'analisi dei soli primi 36 byte di un frame contenente un datagramma IP (in questo caso Linux Cooked +IP header = 36 byte) la funzione pcap_dispatch() recupera i pacchetti e invoca per ciascuno la funzione di analisi processPacket() contenuta in analyzer.c; le ulteriori opzioni che possono essere passate al programma -a e -g servono nel primo caso a scegliere su quali dati basare l'analisi (IP destinazione, sorgente o entrambi) e nel secondo a passare il path del database GeoLite City. Il main carica in memoria il database per ottimizzare i tempi di risposta dell'operazione successiva che vedrà ogni frame del savefile analizzato e passato alla funzione di analisi citata prima (se è conforme al filtro installato). Una volta completata l'analisi e popolata la hash table con tutti i dati che ci interessano il main entra in modalità “web server”, crea un socket sulla porta 3001 e si mette in attesa di una qualsiasi richiesta su questo socket, alla prima richiesta ricevuta si invoca la funzione sendChartPage() (contenuta in webStuff.c) che si occupa dell'invio al browser della pagina html con i risultati, a questo punto il programma termina.

Per poter suddividere le funzioni su più file, così da rendere più modulare il programma, sono state introdotte delle variabili globali, ad esempio: contatori per vari valori considerati interessanti (numero dei pacchetti analizzati, scartati...ecc), puntatore al database aperto, puntatore alla hash popolata e un intero rappresentante il link layer sul quale si è effettuato lo sniffing. Gli header files dei rispettivi sorgenti C permettono di strutturare le funzioni in maniera modulare e di esportare le variabili condivise senza doversi preoccupare della concorrenza, proprio perché il programma ha un solo thread non sono necessari lock o mutex a tutto vantaggio della semplicità. Nel file analizer.c sono contenute le funzioni per l'analisi dei pacchetti, in realtà solo la processPacket() è visibile dall'header le altre funzioni sono statiche e vengono utilizzate interamente per effettuare interrogazioni al database e per popolare la hash con i dati rilevati. La hash table è realizzata utilizzando uthash.h, le funzioni di inserzione, ricerca, cancellazione e ordinamento sono contenute nel file myhash.c