WiresharkListener

Lorena Modica

lorenamodica@gmail.com 30

giugno 2022

Sommario:

1. Introduzione

Wireshark Listener (WL) è un' estensione del software Wireshark il cui obiettivo è quello di avere informazioni riguardo ad un host *X* tramite il suo indirizzo ip (ipv4) . In particolare si vogliono stimare il numero di host diversi contattati da *X*, il numero di volte in cui *X* è stato contattato da host diversi e il numero di pacchetti scambiati con *X*.

La stima è calcolata tramite l' algoritmo HLL (HyperLogLog) che conta elementi distinti in un insieme. Al fine di non sprecare risorse computazionali, l' algoritmo restituisce un valore che non è necessariamente uguale al valore reale della cardinalità dell' insieme in questione.

Questo perché calcolare l' esatta cardinalità di un insieme richiede memoria proporzionale alla dimensione del set , impensabile per insiemi molto grandi.

Si perde in precisione ma si guadagna in performance.

1.1 Prerequisiti

L' esecuzione del tool necessita la presenza di Wireshark sul proprio dispositivo e l'accesso alla rete.

1.2 Informazioni sull' utilizzo

- Aprire WIreshark -> Aiuto -> Informazioni su Wireshark -> Cartelle
- Aprire il path relativo a plugin personali
- Da terminale eseguire il seguente comando per scaricare il contenuto di WL nella cartella dei plugin personali:

```
git clone https://github.com/LorenaModica/wiresharkListener.git
```

- Aprire Wireshark con i permessi di root

1.3 Informazioni utili

Questo codice è implementato su Ubuntu 21.10 con Lua versione 5.2.4 testato su Wireshark versione 3.4.7

2. HLL

HLL è figlio dell' algoritmo Flajolet–Martin (1984) e in generale si basa sull'osservazione che la cardinalità di un multiset *M* di numeri casuali uniformemente distribuiti può essere calcolata tramite il numero di leading zeros[1] nella rappresentazione binaria di ogni numero nell' insieme.

Se il massimo numero di leading zeros osservati è *n*, una stima per il numero di elementi distinti nell ' insieme e 2^n

HLL richiede l' applicazione di una funzione hash ad ogni elemento dell' insieme iniziale per ottenere il multiset *M,* che avrà uguale cardinalità del set iniziale. La stima della cardinalità fatta da HLL viene calcolata su *M*.

2.1 Operazioni

Nell' implementazione di WL , Hyperloglog contiene , tra le operazioni ovvie di inizializzazione, *reset* e *destroy* anche quelle di *add* e *count*.

2.1.1 hll_add (hll , item)

Questa operazione prende in input una table[2] e calcola una funzione hash sulla stringa item.

Prima di andare nel dettaglio delle funzioni ausiliarie utilizzate dall' operazione sopracitata, è doveroso fare un breve accenno sulla funzione hash scelta: SHA256.

L' implementazione in Lua di SHA256 è stata reperita su una repo pubblica GitHub, il cui link è riportato sul codice sorgente della stessa.

---

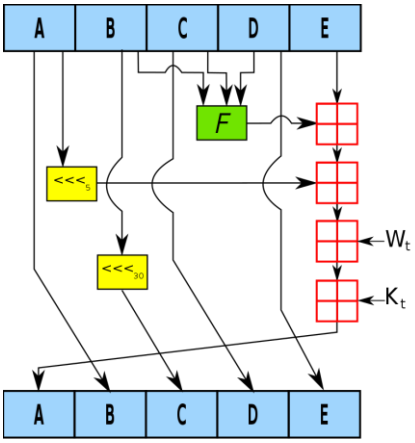
[1] Leading zeros : è il numero di zeri iniziali che precedono un valore significativo , ad esempio 007 ha 2 leading zeros [2] Table:

unica e principale struttura dati nativa del linguaggio Lua

SHA è una famiglia di funzioni hash che trova largo impiego nella crittografia.

Il messaggio originale passato a una delle funzioni SHA, tramite alcune operazioni di padding, shifting di bit e ausilio di funzioni non lineari stravolgono l'input iniziale.

Il fulcro delle SHA è la funzione di compressione.



Per maggiore completezza e chiarezza, in seguito, una figura che rappresenta un'iterazione all'interno della funzione di compressione di SHA1

A, B, C, D ed E sono parole di stato a 32 bit; F è una funzione non lineare che varia; $\lll_n$ denota una rotazione del bit di sinistra di n posti; n varia per ogni operazione. $\boxplus$ denota l'addizione modulo 2^{22}. K_t è una costante.

Fonte:Secure Hash Algorithm - Wikipedia

Per concludere, SHA256 è la versione migliorata di SHA1 a 256 bit.

Tornando ad *hll_add* , dopo aver generato l' hash sulla stringa passata come parametro, viene calcolato e salvato il valore massimo di *leading zeros* relativi a quell' hash generato.

2.1.2 hll_count (hll , item)

In questa funzione , in base alla precisione scelta[3] e agli zeri calcolati precedentemente viene effettuata la "manovra matematica". Il discorso è molto più complesso perché in realtà viene calcolata la media armonica, vengono usate delle costanti e delle funzioni matematiche note.

[2] Precisione scelta: quando viene inizializzata una hll table viene scelto il grado di precisione, cioè il numero di bit per rappresentare i bucket relativi ai numeri. Più il numero dei bit è alto più aumenta la precisione della stima.

3.Test

Utilizzando come dati di riferimento i valori delle statistiche di Wireshark, questi sono i risultati approssimati di WL impostando il numero di bit a 19.

Le informazioni di seguito fanno riferimento all' indirizzo 192.168.1.X

- Valori reali

Ethernet · 10 | IPv4 · 10 | IPv6 · 5 | TCP · 2 | UDP · 15

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A
35.232.111.[redacted]	192.168.1.[redacted]	10	911	5	486	5
185.125.188.[redacted]	192.168.1.[redacted]	20	11k	12	8.044	8
185.125.190.[redacted]	192.168.1.[redacted]	2	180	1	90	1
192.168.1.[redacted]	224.0.0.1	1	60	1	60	0
192.168.1.[redacted]	192.168.1.[redacted]	6	648	3	348	3
192.168.1.[redacted]	224.0.0.[redacted]	2	133	2	133	0
192.168.1.[redacted]	224.0.0.[redacted]	31	9.363	31	9.363	0
192.168.1.[redacted]	239.255.255.[redacted]	13	2.719	13	2.719	0
192.168.1.[redacted]	192.168.1.[redacted]	5	430	5	430	0
192.168.1.[redacted]	224.0.0.[redacted]	1	87	1	87	0

- Valori approssimati

4. Considerazioni finali

Si potrebbe pensare che HLL si comporti poco bene nei casi in cui la funzione hash scelta generi un alto numero di collisioni. In realtà nel calcolo della stima viene considerata anche questa eventualità, cercando di ridurre al minimo errori dovuti alle collisioni.

4.1 Uso pratico

Questo tool può essere utile per intercettare attacchi DDoS proprio perché si appoggia su HLL.

In termini di performance non appesantisce ulteriormente la macchina se vi è in atto un attacco DDoS ed inoltre riesce a fare delle stime veloci che possono darci l' idea di un' eventuale anomalia.