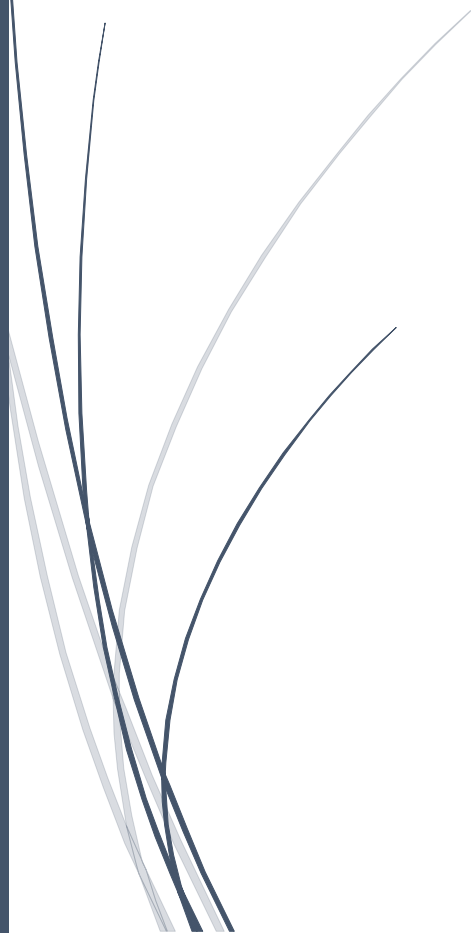
A.A. 2018/19 – Corso di Gestione di Reti

WESERA

Web Server Redis Analyzer



Mattia Dui
MAT. 545577

WESERA è un piccolo tool in Python 3.x creato per l'analisi dell'efficacia di Redis come caching in una installazione Web Server (Apache, Ngnix, ecc.) + MySQL/MariaDB. Nello specifico il tool è stato pensato per un suo utilizzo come metrica di efficacia all'interno di una installazione Wordpress e MySQL con caching degli oggetti (articoli, pagine interne, ecc.) su Redis.

Implementazione

Il tool analizza i pacchetti da e per il server Redis sulla interfaccia locale della macchina (ma può essere scelta una qualsiasi interfaccia come "locale" attraverso l'argomento opzionale relativo) e classifica le richieste in "set" (comando Redis SET), "hit" (comando Redis GET) o "admin" (altri comandi Redis), guardando il payload dei pacchetti verso il server Redis, e visualizza anche il rapporto tra numero di "hit" e "set". Dato che il tool è stato sviluppato per analisi dell'efficacia del caching di un blog, si assume che ogni richiesta GET sia effettuata con l'assunzione di presenza della chiave e/o previa verifica mediante il comando EXISTS. Infatti, Redis restituisce NULL (nil) nel caso la chiave non esista ma anche nel caso che il valore associato alla chiave sia NULL.

L'analisi del payload si basa sulla struttura molto semplice dei pacchetti inviati dal client a riga di comando redis-cli: oltre ad alcuni byte di delimitazione, il payload è esattamente uguale alla richiesta inoltrata con formato <comando> <valori>; pertanto il tool controlla la posizione della parola/comando "GET" o "SET" e determina il tipo di richiesta inviata al server in base a ciò. Poiché Redis non pone limiti alle parole utilizzabili come chiavi, una richiesta "GET SET" o "SET GET GET" è valida. Ne consegue che va determinata quale è la prima parola (intesa come parola di linguistica, non come parola di bit) della richiesta per poterla classificare.

Per quanto riguarda i pacchetti in uscita dal server Redis, il tool quantifica il volume di dati trasmessi per avere una metrica aggiuntiva e, sempre per questo motivo, viene anche quantificato il volume di dati recuperati dal server MySQL qualora quest'ultimo utilizzi l'interfaccia di rete in loopback come Redis e non usi invece il socket UNIX nativo.

Il tool effettua lo sniffing dei pacchetti utilizzando Pyshark con il seguente filtro BPF:
`tcp port 6379 or tcp src port 3306`
dove i numeri di porte possono essere cambiati attraverso le opzioni relative.

Il tool visualizza le informazioni in stdout sul terminale ogni 5 secondi attraverso un thread parallelo non bloccante sullo sniffing, facendo un campionamento in base ai dati attuali che il packet sniffer ha accumulato in una variabile condivisa.

Inoltre, il tool permette anche di salvare le informazioni in un report dove ogni 2 minuti vengono salvate le metriche rilevate. Anche questa operazione si svolge in un thread parallelo per non bloccare lo sniffing dei pacchetti.

Dipendenze

- Pyshark
  `pip3 install pyshark`
  Per lo sniffing e l'ispezione dei pacchetti
- Timeloop
  `pip3 install timeloop`
  Per l'esecuzione dei thread paralleli ciclici

Versione di Python richiesta: > 3.5.x
Versione di Python consigliata: > 3.6.x

Testato su Ubuntu 18.04 + Python 3.6.5 , Debian 9 + Python 3.5.3

Nota: se eseguito su Python 3.5.x alla chiusura di WESERA mediante CTRL+C potrebbe essere sollevato un errore di asyncio. Tuttavia, suddetto errore non compare nell'esecuzione su Python 3.6.

Esecuzione & Opzioni

Il tool deve essere eseguito da utente che abbia diritti superuser oppure da utente root. Su Ubuntu ad esempio:

```
sudo python3 wesera.py [eventuali opzioni]
```

Di seguito le opzioni disponibili:

`-h, --help`
Visualizza l'aiuto in inglese

`-i LOCAL_INTERFACE, --interface LOCAL_INTERFACE`
Nome dell'interfaccia dove effettuare lo sniffing dei pacchetti [default = lo]

`-pr REDIS_PORT, --port-redis REDIS_PORT`
Porta del server Redis [default = 6379]

`-pm MYSQL_PORT, --port-mysql MYSQL_PORT`
Porta del server MySQL [default = 3306]

`-s, --silent`
Non visualizza le informazioni sul monitor stdout (Necessita dell'opzione -r ; consigliata per l'utilizzo in background con nohup)

`-r, --report`
Genera un report file aggiornato ogni due minuti (Necessaria per usare -rn)

`-rn PATH_TO_FILE, --report-name PATH_TO_FILE`
Percorso assoluto al file di report, completo di nome del file [default = wesera_report.txt]