Corso di Laurea in Tecnologie Informatiche

Corso di Gestione Reti

Anno Accademico 2009/2010

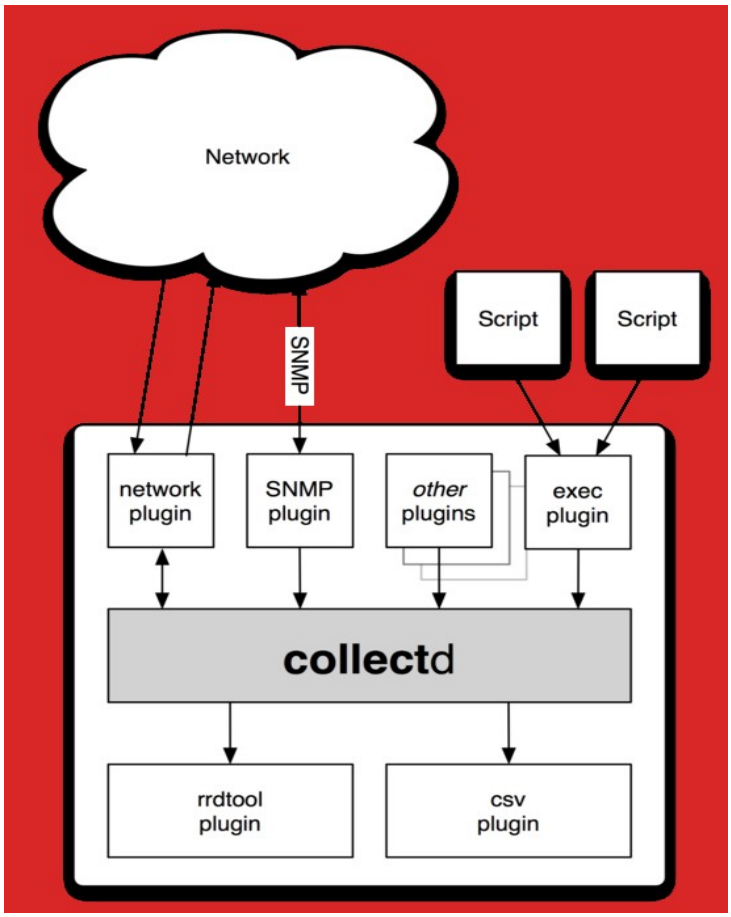
PLUGIN PER IL MONITORAGGIO DEL TRAFFICO HTTP PER COLLECT

Daniele Sirigu

INDICE

1. INTRODUZIONE

In questa relazione verranno presentate le caratteristiche e il funzionamento di un plugin per l'analisi del traffico HTTP per collectd.
Collectd (http://www.collectd.org) è un demone che raccoglie periodicamente informazioni sul sistema in cui è installato e fornisce meccanismi per immagazzinare i dati in vari formati, per esempio in files RRD.



In collectd, qualsiasi cosa ad eccezione del parsing del file di configurazione, è realizzata tramite plugin. Questo implica che il demone principale non ha nessuna dipendendenza esterna e dunque può essere eseguito su qualsiasi sistema POSIX mentre al contrario, i plugins ovviamente possono averne.

I plugin possono essere scritti in C, Java, Python ed altri linguaggi. Quelli in C hanno il vantaggio di essere caricati direttamente dal demone, il che garantisce migliori prestazioni rispetto alle altre soluzioni.

Vi sono due gruppi principali di plugin: quelli d'input e quelli d'output. I primi, con una certa frequenza, acquisiscono in qualche modo il valore corrente e lo inviano al demone. I secondi recuperano dal demone i valori trasmessi dagli

altri plugin ed eseguono un qualche genere di elaborazione (scrittura su file RRD, CSV o altro)

Il plugin realizzato utilizza la libreria libpcap () per la cattura di tutto il traffico sulla porta 80.
E' possibile impostare l'interfaccia su cui il plugin andrà ad operare ed inoltre lo si può configurare in modo da ignorare i pacchetti inviati dal proprio host.

Le metriche gestite dal plugin sono:
-Il numero totale di ottetti relativo al traffico HTTP
-Il numero di pacchetti con un certo tipo di Request Method (GET, POST, ..)
-Il numero di pacchetti con un certo tipo di Status Code (100, 200, 201, ..)
-Il numero di pacchetti con un certo Mime Type (text,image, ..)

2. INSTALLAZIONE E INTEGRAZIONE CON COLLECTD

Il plugin è stato testato sul sistema operativo Ubuntu 10.4 e Debian squeeze con collectd versione 4.10.0.

Il primo passo per l'integrazione del plugin è quello di scaricare la versione di collectd per sviluppatori:

git clone git://git.verplant.org/collectd.git

Una volta clonato il repository, si devono copiare i seguenti files:
http.c utils_http.h utils_http.c collectd.conf.in types.db Makefile.am
all'interno della cartella src mentre il file *Configure.in* va nella directory principale di collectd.

Collectd utilizza automake e autoconf per creare un Makefile. Per consentire la compilazione del plugin sono stati dunque modificati i files: *Configure.in* e *Makefile.am*.
E' stato modificato anche il file *collectd.conf.in* in modo tale che il plugin in questione sia lanciato assieme al demone.

Il plugin, in aggiunta alle librerie richieste da collectd, necessita della libreria libpcap (http://www.tcpdump.org). Una volta installata, si può eseguire lo script *build.sh* per la generazione dei files relativi ad automake e autoconf e poi configurare, compilare ed installare il tutto con la sequenza di comandi:
./configure
make
sudo make install

Di default, collectd viene installato in /opt/collectd e l'eseguibile si trova

all'interno della directory */sbin* (collectd).

Per visualizzare l'output si può usare un plugin come RRDtool e dai files RRD si possono generare dei grafici. Per la generazione di questi ultimi, durante i test è stato utilizzato lo script cgi incluso nella directory *contrib/collection3* di collectd.

3. ARCHITETTURA

Una volta che un plugin viene caricato in memoria, il demone chiama la funzione void *module_register(void);* per registrare una o più callbacks.

Nel caso del plugin in questione la funzione è la seguente (in *http.c*):

```
void module_register (void)
{
        plugin_register_config ("http", http_config, config_keys,
config_keys_num);
        plugin_register_init ("http", http_init);
        plugin_register_read ("http", http_read);
}
```

La funzione *http_config* viene chiamata se si vuole utilizzare una determinata configurazione e viene invocata per ogni opzione selezionata (l'interfaccia su cui il plugin andrà ad operare o la possibilità di poter ignorare i pacchetti inviati dal proprio host).

La funzione *http_init* viene invocata per inizializzare il plugin. Essa viene invocata dopo che sono state eseguite tutte le chiamate alla funzione *http_config*.

Al suo interno viene creato un nuovo thread, che facendo uso della libreria *pcap*, si occupa di intercettare i pacchetti in transito sulla porta 80 e, ogni volta che un pacchetto soddisfa i requisiti dettati dal filtro utilizzato (porta 80), viene chiamata una determinata funzione che si occupa di analizzare il pacchetto. Quest'ultimo può contenere un header HTTP o no. In entrambi i casi si tiene traccia della dimensione del pacchetto.

Nel caso di un pacchetto contenente un header HTTP, si determina se si tratta di una richiesta o di una risposta. Nel primo caso si tiene traccia del tipo di Request Method usato (GET, POST, PUT ecc.) mentre nel secondo dello Status Code (200,301,302 ecc.) e se presente anche del MIME TYPE (text,image, application ecc.). Buona parte di questo processo d'analisi viene svolto tramite delle funzioni presenti in *utils_http.h* e *utils_http.c.*

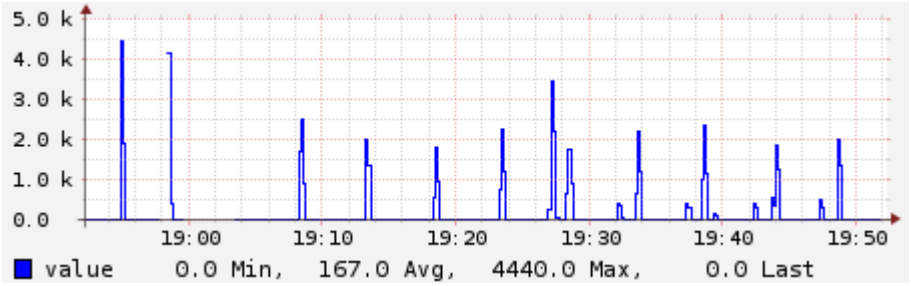
La funzione *http_read* è una reading callback: essa viene chiamata periodicamente (una volta per ogni intervallo di tempo) e si occupa di passare i dati acquisiti a collectd.

4. TESTS E OUTPUTS

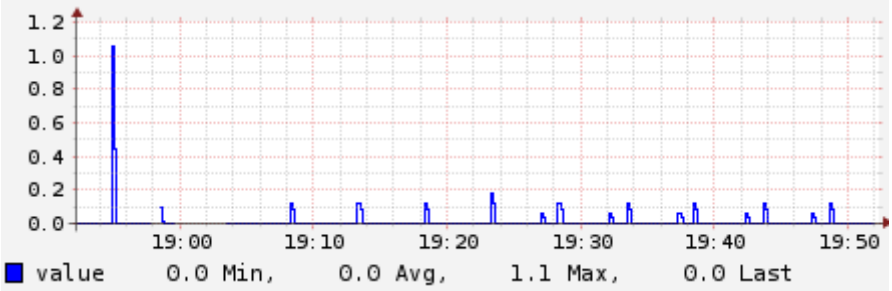
Come si è affermato in precedenza, il plugin è stato testato sia su Debian squeeze sia su Ubuntu 10.04.

Per visualizzare tutti i messaggi d'output è necessario compilare collectd attivando la modalità di debugging (usando l'opzione *–enable-debug* quando si esegue il *./configure); s*i deve inoltre attivare il plugin *logfile* modificando il file di configurazione di collectd (*/opt/collectd/etc/collectd.conf se collectd è stato installato in /opt/*).

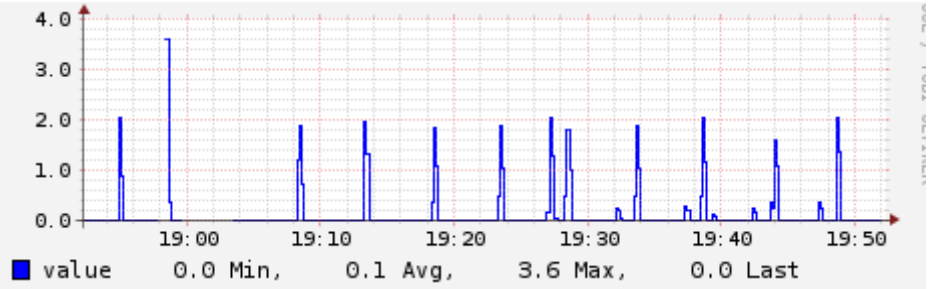
Di seguito, a titolo d'esempio, sono riportati alcuni grafici relativi all'applicazione del plugin, ottenuti per mezzo dello script presente nell'archivio contenente collectd (collection3).



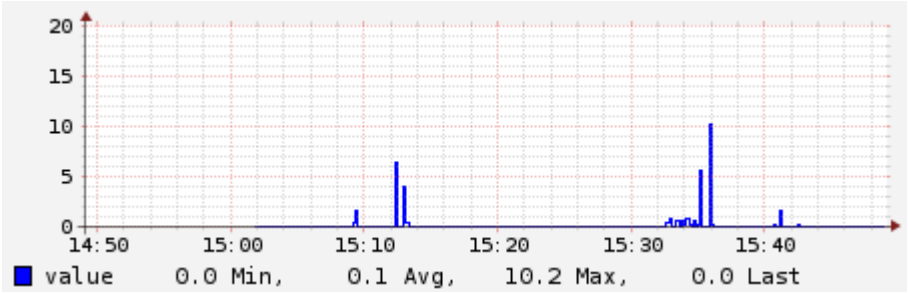
HTTP OCTECTS



MIME TIPE: APPLICATION



REQUEST METHOD: GET



STATUS CODE: 200

Sull'asse delle ordinate vi è indicato il numero di pacchetti con un determinato codice/stato/tipo al secondo (o di ottetti nel primo grafico). L'asse delle ascisse invece rappresenta il tempo. Con lo script utilizzato, è possibile visualizzare statistiche relative a una determinata ora,giorno, settimana o anno.