Progetto Gestione Reti

Antonio Pandolfi
Antonio Di Tommaso

Giugno 2014

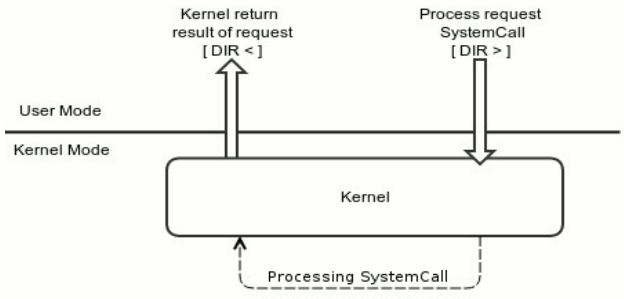
1. Introduzione

Per il progetto finale di Gestione di Reti, abbiamo deciso di implementare alcuni chisels per lo strumento di monitoraggio di sistema SYSDIG (www.sysdig.org) .
Sysdig permette la cattura di tutte le system call effettuate dai processi, in modalità kernel. Una volta catturate permette di filtrare, ed analizzarle, e proprio su questi due ultimi punti che abbiamo implementato i chisels, che descriveremo successivamente, scripts sviluppati in LUA (www.lua.org).
Per l'utlizzo dei chisels è necessario invocare da terminale sysdig con l'opzione -c, e nome del chisel.

```
~$ sysdig -c nome_chisel
```

Per ogni chiamata di sistema effettuata da un processo, abbiamo due fasi, una di domanda, effettuata appunto dal processo verso il kernel, e una di risposta dal kernel verso il processo. Le due fasi, durante la cattura sono identificate rispettivamente dai simboli ' > ' e ' < ' . E' possibile quindi fare un monitoraggio in uno dei due versi, oppure in entrambi i versi (per ogni system call però ci saranno due linee di output).

```
~$ sysdig evt.type=clone
...
238290 12:36:43.158755455 1 compiz (1965) > clone
238293 12:36:43.164016948 1 compiz (1965) < clone res=3524(compiz) exe=compiz args=
tid=1965(compiz) pid=1965(compiz) ptid=1852(gnome-session)
...
```



2. Chisels

2.1 sysdig -c **httpCounter**

httpCounter è un chisel sviluppato al fine di monitorare le richieste http, di tipo GET, effettuate dai vari processi.
Vengono filtrate le system call che coinvolgono i socket aperte sulla porta 80.
Per ogni processo, durante la fase di monitoraggio, viene restituito l'URL e a fine esecuzione viene restituito il numero di richieste effettuate.

2.2 sysdig -c **memoryBrk**

memoryBrk, invece riguarda il monitoraggio della memory allocata dai processi, durante la loro esecuzione.
Vengono filtrate le system call riguardanti le richieste di memoria fatte dai processi. Per ogni processo, durante la fase di monitoraggio, vengono conteggiati e sommati i valori di ritorno dal kernel.

2.3 sysdig -c **forkRate [timout]**

questo chisel conteggia quanti processi figli vengono generati dai processi già in esecuzione. Sono state monitorate le system call clone ed execve, e raggruppate per pid del processo chiamante.
L'output è generato ad intervalli periodici in base al timout passato come argomento all'invocazione.

2.4 sysdig -c **forkRateTab**

questo chisel è una variante di quello precedente, nel quale però viene utilizzato la chiamata alla funzione table_generator, implementata dagli sviluppatori di sysdig e l'intervallo di output è quello di default. La funzione rable_generator permette, oltre che a una più semplice gestione delle tabelle di output, anche di ottenere già i dati in formato JSON.