Sysdig – administratorov švicarski nožić

Zar još jedan u nizu alata za dojavu stanja sustava? Bolji Emacs od Emacsa? Zbilja, taj ***sysdig***... zar bismo zbog njega trebali napustiti stare i iskušane alate?

Sysdig je u odnosu na staru gardu aplikacija slične namjene uistinu novak, no mladac je uistinu darovit. Kao i njegovi kolege po zadaći, ***sysdig*** se sastoji od kernel modula kojeg je potrebno "ugurati" u jezgru i *user space* aplikacije koja prikuplja željene podatke. Sustav može nadzirati u stvarnom vremenu, ili pak može prikupljati i u log zapisivati željene parametre sustava za kasniju obradu.

Po čemu se, onda, razlikuje novopridošlica? Kako se autori vole hvaliti, kernel modul *sysdiga* napisan je da što manje opterećuje operacijski sustav i zato sadrži samo najnužniju funkcionalnost, dok su više moždane funkcije (sortiranje, filtriranje, skripte...) smještene u korisnički prostor. To naravno nije garancija da u neka specifična operacija neće značajno opteretiti resurse, što se spominje i u primjerima: autori preporučuju da intenzivnije operacije koje skupljaju mnoštvo podataka ipak ne pokrećete reda radi, već onda kad vam je u ciljniku specifičan problem sa sustavom. Opterećenje se najviše osjeti korištenjem možda najvrijednije sposobnosti ovog softvera: izrade "snapshot" log datoteke iz koje je naknadno moguće izvući mnoštvo podataka o ponašanju sustava u određeno vrijeme.

Prava zabava počinje na korisničkoj strani. Čak i ako zaboravimo na mogućnost izrade skripti i korištenje gotovih biblioteka, mogućnosti čeprkanja po podacima su zaista brojne.

Najjednostavnija i najmanje korisna (ali zato spektakularna) naredba je jednostavno pozivanje programa *sysdig* u konzoli bez parametara: *sysdig* će ispisivati sve sistemske događaje u stilu *strace* naredbe; izvrsno kad želite impresionirati nekoga svojim tajanstvenim i dubokim znanjima. Imajte na umu da *sysdig* morate pokrenuti kao administrator ili ga dodati u odgovarajuću grupu i /etc/sudoers datoteku.

Format ispisa je jednostavan, svaki događaj ispisuje se u jednom retku, na propisan način (ako ga sami ne promjenite korištenjem argumenta "-p"), a ispis sirovih podataka se ograničava na prvih 80 bajtova (ako ne izaberete drugu vrijednost, naravno):

```
%evt.num %evt.time %evt.cpu %proc.name (%thread.tid) %evt.dir %evt.type %evt.args
```

Za razliku od *strace* programa, *sysdig* razlikuje ulazne i izlazne događaje, te raspisuje (mrežne i datotečne) deskriptore u čovjeku čitljivijem obliku.

Naravno, kompletna funkcionalnost aplikacije tog tipa zahtjeva spremanje događaja u neku datoteku, što ćete postići parametrom -w:

```
sysdig -w mojlog.scap
```

dobivenu datoteku zatim možete čitati sa:

```
sysdig -r mojlog.scap
```

Želite li podatke spremiti u čovjekočitljivom obliku, koristite parametar -A:

```
sysdig -A > mojlog.txt
```

Prava vrijednost ovog programa su filteri kojima iz silne gomile podataka izvlačite ono što želite (bilo zadavanjem filtera kao realtime opcije ili kroz čitanje već pospremljene .scap datoteke).

Popis filtera u instaliranoj verziji sysdig aplikacije možete dobiti naredbom:

```
sysdig -L
```

dok

```
sysdig -l
```

daje popis polja po kojima je moguće filtrirati ispis.

Najbolji početak igranja sa ovom moćnom alatkom je korištenje tzv. "***chisela***", gotovih skripti za prikaz raznovrsnih podataka napisanih u programskom jeziku ***Lua***.

Krenimo odmah u glavu: pokrenimo *chisel* koji špijunira korisnike na serveru:

```
kanta rado # sysdig -c spy_users
9331 14:05:55 rado) dircolors --print-database
9331 14:05:55 rado) dircolors
9331 14:05:56 rado) ls /etc/bash_completion.d
9331 14:05:56 rado) /bin/bash /usr/bin/mint-fortune
9331 14:05:56 rado) gsettings get com.linuxmint.terminal show-fortunes
```

U gornjem primjeru *sysdig* je uredno "otkucao" root korisniku sekvencu naredbi koje su se izvršile mojim jednostavnim klikom miša na ikonicu terminal emulatora.

Naravno, korisnike je moguće špijunirati na različite načine, pregledavati koje datoteke otvaraju i slične nepodopštine, no ovaj alat je prije svega zamišljen za uočavanje sistemskih, a ne korisničkih problema.

Tako je *chisel* nazvan *bottleneck* zadužen da ispiše top listu najsporijih procesa koje je pronašao u log datoteci; naravno, *chisel* možete pokrenuti i "na živo", ali on neće ispisati nikakve statističke podatke dok ne prekinete izvršavanje kombinacijom tipki Ctrl-C:

```
sysdig -c bottlenecks
5) 0.000000000 rs:main (856) > futex addr=811044 op=128(FUTEX_PRIVATE_FLAG) val=729
155) 0.003999223 rs:main (856) < futex res=0
8456) 0.000000000 configmgrWriter (13021) > rename
8469) 0.000181409 configmgrWriter (13021) < rename
3098) 0.000000000 gkrellm (10007) > read fd=5(<f>/proc/diskstats) size=1024
3099) 0.000108539 gkrellm (10007) < read res=1024 data= 1 0 ram0 0 0 0 0 0 0 0 0 0
0. 1 1 ram1 0 0 0 0 0 0 0 0 0 0.
6715) 0.000000000 gkrellm (10007) > read fd=5(<f>/proc/diskstats) size=1024
6716) 0.000107362 gkrellm (10007) < read res=1024 data= 1 0 ram0 0 0 0 0 0 0 0 0 0
0. 1 1 ram1 0 0 0 0 0 0 0 0 0 0.
2644) 0.000000000 soffice.bin (10219) > rename
2741) 0.000097465 soffice.bin (10219) < rename
6705) 0.000000000 gkrellm (10007) > read fd=11(<f>/proc/stat) size=1024
6706) 0.000066126 gkrellm (10007) < read res=1024 data=cpu 326801 3 65725 7599463 476
```

```
93 3852 0 0 0 0.cpu0 55985 2 14513 904364 27297 2
3088) 0.000000000 gkrellm (10007) > read fd=11(<f>/proc/stat) size=1024
3089) 0.000064844 gkrellm (10007) < read res=1024 data=cpu 326794 3 65724 7599216 476
93 3852 0 0 0 0.cpu0 55984 2 14513 904333 27297 2
2310) 0.000000000 soffice.bin (10219) > open
2315) 0.000048526 soffice.bin (10219) < open fd=32(<f>/home/rado/.local/share/recentl
y-used.xbel.KITXJX) name=/home/rado/.local/share/recently-
used.xbel.KITXJX flags=39(O_EXCL|O_CREAT|O_RDWR) mode=0
```

Chisele je moguće dodatno filtrirati standardnim filterima, pa ako želimo izbjeći ponavljanje dosadnih informacija o procesima *gkrellm* i *nvidia-settings* koristit ćemo filter "proc.name!=":

```
sysdig -c bottlenecks proc.name!=gkrellm proc.name!=nvidia-settings
44952) 0.000000000 Chrome_IOThread (10909) > write fd=251(<u>) size=40
44954) 0.001839030 Chrome_IOThread (10909) < write res=40 data=........2.,..<.......J
..................
71320) 0.000000000 irqbalance (1351) > read fd=3(<f>/proc/interrupts) size=1024
71321) 0.000067553 irqbalance (1351) < read res=1024 data= 0 0 0 0 0 0 0 PCI
71332) 0.000000000 irqbalance (1351) > read fd=3(<f>/proc/stat) size=1024
71336) 0.000059708 irqbalance (1351) < read res=1024 data=cpu 335095 3 67901 7972487
48392 3938 0 0 0 0.cpu0 57629 2 14956 950202 27333 2
46596) 0.000000000 SGI_video_sync (10985) > ioctl fd=9(<f>/dev/nvidiactl) request=C02
0462A
46654) 0.000042004 SGI_video_sync (10985) < ioctl res=0
71316) 0.000000000 irqbalance (1351) > read fd=3(<f>/proc/interrupts) size=1024
71317) 0.000040338 irqbalance (1351) < read res=1024 data= CPU0 CPU1 CPU2 CPU3 CPU4 C
PU5 CP
47471) 0.000000000 chrome (10912) > clock_gettime
47473) 0.000039456 chrome (10912) < clock_gettime
47362) 0.000000000 SGI_video_sync (10985) > ioctl fd=9(<f>/dev/nvidiactl) request=C02
0462A
47363) 0.000031358 SGI_video_sync (10985) < ioctl res=0
46008) 0.000000000 chrome (10880) > madvise
46009) 0.000028147 chrome (10880) < madvise
```

Možete koristiti sljedeće komparatore:

```
= jednako
!= nije jednako
> ve?e od
>= ve?e ili jednako
< manje
<= manje ili jednako
```

contains argument sadrži navedeni niz znakova

Za spajanje komparatora koristite klasične Boolove izraze *and*, *or* i *not*, a također je dozvoljeno i korištenje zagrada.

Jedan od zgodnijih *chisela* je i onaj za provjeru mrežnog prometa prema određenom serveru: jednom pokrenut, pratit će mrežni promet:

```
sysdig -A -c echo_fds fd.sip=161.53.160.81
------ Write 528B to 192.168.1.73:58732->161.53.160.81:80
GET / HTTP/1.1
Host: sistemac.carnet.hr
```

```
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ch
rome/36.0.1985.125 Safari/537.36
DNT: 1
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-GB,en;q=0.8,en-US;q=0.6,hr;q=0.4
Cookie: SESS65244fa1ede9cfa2aec8b51f633b0f93=8jbopunbfpulo02iid55nhp9a4; has_js=1
If-Modified-Since: Fri, 25 Jul 2014 13:12:03 GMT
------ Read 2.61KB from 192.168.1.73:58732->161.53.160.81:80
HTTP/1.1 200 OK
Date: Fri, 25 Jul 2014 13:12:18 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7+squeeze19
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Last-Modified: Fri, 25 Jul 2014 13:12:18 GMT
Cache-Control: store, no-cache, must-revalidate
Cache-Control: post-check=0, pre-check=0
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 7453
Keep-Alive: timeout=15, max=96
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8
][o9~tK-tNN;8ItaPUDXM*fLe<?2d!JcQ:\\Y=@9|{w{tozFvTZfU{t1tU_5A, P*h;Q
\\\\Z5Bv@6dfvB<{wvT*$lV:`f!uBjCs6Jx
x4\"w*4LFTC^
```

Uočite kako smo koristili opciju "-A" da bi prikaz rezultata bio čitljiv. Nije *Wireshark*, ali je sasvim upotrebljivo kad trebate "uhvatiti" i analizirati problematične konekcije.

Korištenjem već gotovih *chisela* izbjegavate stvaranje kobasica u naredbenom retku i dobivate lijepo uobličen ispis raznih *quick&dirty* informacija: o najvećem zauzeću procesora, mrežnih portova, najkorištenijih sistemskih poziva... u načelu ništa što ne biste mogli dobiti i nekim drugim alatom ili kombinacijom nekoliko drugih alata.

Stvar postaje daleko zanimljivija i praktičnija korištenjem filtera u suradnji sa *chiselima*, no pritom valja biti oprezan jer ne prihvaćaju svi *chiseli* sve filtere, a neki od nih zahtjevaju i određene argumente.

Ako vas previše informacija na zaslonu zbunjuje, sysdig može formatirati ispis onako kako vama najbolje odgovara – korištenjem opcije -p iza koje slijedi unutar navodnika definiran način prikaza, a kojeg također formirate korištenjem istih onih filtera koje koristite u samoj aplikaciji, pa će primjerice:

```
sysdig -p "Program: %proc.name Port: %fd.cport" fd.l4proto=udp
```

daje ispis aplikacija i portova preko kojih sa drugom stranom komuniciraju UDP protokolom.

Trebate li zamjeniti svoje stare iskušane programe ovim mladcem? Naravno da ne – ma koliko *sysdig* bio zanimljiv, obećavajući i već sad upotrebljiv, riječ je o vrlo svježem projektu koji tek treba doseći svoju pravu zrelost. Mudro je zato strpiti se, ali i pozorno pratiti ovog novaka – jer obećava zaista mnogo.

Ako još niste uvjereni u simpatične mogućnosti ovog programa, pročitajte dvije priče o lovu na *crackere*:

Fishing for Hackers: Analysis of a Linux Server Attack [1]

Fishing for Hackers (Part 2): Quickly Identify Suspicious Activity With Sysdig [2]

uto, 2014-08-05 16:46 - Radoslav Dejanović**Kuharice:** Linux [3]
Kategorije: Software [4]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr./node/1423

Links
[1] http://draios.com/fishing-for-hackers/
[2] http://draios.com/fishing-for-hackers-part-2/
[3] https://sysportal.carnet.hr./taxonomy/term/17
[4] https://sysportal.carnet.hr./taxonomy/term/25