SystemTap Tutorial, Part 1

This is the first of a two-part series on SystemTap, a dynamic method to monitor and trace the operations of a running Linux kernel. SystemTap is useful to systems administrators, kernel developers, support engineers, researchers and students.

“Who is doing the maximum read/write operations on my server?”
“Can I add some debug statements in the kernel without rebuilding it and rebooting the system?”

These are questions you might have asked yourself, if you are a systems administrator or a kernel developer. Let’s see what options are available in answer to these questions:

- **Tracing:** Provides information while running, and gives a quick overview of code flow, but gives a lot of information. Tools like strace, ltrace and ftrace (http://www.opensourceforu.com/2010/11/kernel-tracing-with-ftrace-part-1/) are used for tracing.
- Profiling: Does sampling while running, and we can do the analysis after the event. Oprofile is used for sampling.
- Debugging: We can set breakpoints, look at variables, memory, registers, stack trace, etc. We can debug only one program at a time, and the debugger stops the program while we do the inspection. GDB (http://www.opensourceforu.com/2011/11/gdb-logging-function-parameters-part-1/)/KDB is used for such debugging.

So which of these tools would you use? You’re probably thinking of using a combination of them; wouldn’t it be great to have the capabilities of all these tools combined into one? The response to just such a wish is, SystemTap!

Welcome to SystemTap!

SystemTap can monitor multiple system-wide synchronous and asynchronous events at the same time. It can do scriptable filtering and statistics collection. It’s a dynamic method of monitoring and tracing the operations of a running Linux kernel.

To instrument the running kernel, SystemTap uses Kprobes (http://www.opensourceforu.com/2011/04/kernel-debugging-using-kprobe-and-jprobe/) and return probes. With kernel debug information, it gets the addresses for functions and variables referenced in the script. With utrace, SystemTap supports probing user-space executables and shared libraries as well. SystemTap is, therefore, useful to systems administrators, kernel developers, support engineers, researchers and students.

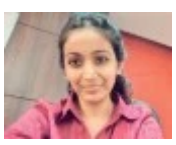
Installation

To install SystemTap on Fedora, run the following commands as root:

```
yum install systemtap kernel-devel
debuginfo-install kernel
```

016/03/we-have-not-faced-any-glitches-as-we-do-the-security-hardening%c2%94/)
DIKSHA P GUPTA , ...

(http://opensourceforu.com/2016/01/%c2%93from-a-personal-preference-perspective-i-would-choose-an-open-source-solution-any-day/)
From a personal preference perspective, I would choose an open source solution, any day (http://opensourceforu.com/2016/01/%c2%93from-a-personal-preference-perspective-i-would-choose-an-open-source-solution-any-day/)
DIKSHA P GUPTA , ...



(http://opensourceforu.com/2015/11/open-source-technology-first-love-of-the-leaders-at-askmebazaar-com/)
Open Source Technology: First Love of the Leaders at AskMeBazaar.com! (http://opensourceforu.com/2015/11/open-source-technology-first-love-of-the-leaders-at-askmebazaar-com/)
DIKSHA P GUPTA , ...

(http://opensourceforu.com/2015/08/open-source-and-problem-solving-attitude-is-the-way-to-work-at-zopper/)

To use SystemTap on Ubuntu or any other distro, you need to install the `systemtap` package, and the `debuginfo` packages corresponding to the kernel you're running.

You need to be the root user to run the SystemTap scripts — or you could add a normal user account to either the *stapdev* or *stapusr* groups, to allow that account to run the script.

How does it work?

To understand how SystemTap works, let's run a script in verbose mode (with the `-v` switch). The `stap` program is the front-end to SystemTap. The `-e` switch instructs it to execute the script in the following argument:

```
$ stap -v -e 'probe syscall.read {printf("syscall %s arguments %s \n",
Pass 1: parsed user script and 65 library script(s) using 83596virt/204
Pass 2: analyzed script: 1 probe(s), 4 function(s), 0 embed(s), 0 globa
Pass 3: translated to C into "/tmp/stapUGVeZi/stap_b40c8268c87acc683f75
Pass 4: compiled C into "stap_b40c8268c87acc683f75ded62a52ee66_2113.ko"
Pass 5: starting run.
syscall read arguments 4, 0x00007fffa773b4c0, 8196
Pass 5: run completed in 20usr/60sys/174real ms.
```

Let's see what happened at each of the passes mentioned:

- Passes 1 and 2: The script we want to run is parsed, and the code is checked for semantic and syntactic errors. Any tapset reference is imported. Debug data (provided via `debuginfo` packages) is read to find the addresses for functions and variables referenced in the script.
- Pass 3: The script is translated into C code.
- Pass 4: The translated C code is compiled to create a kernel module.
- Pass 5: The compiled module is inserted into the running kernel.

Once the module is loaded, probes are inserted at proper locations. From now on, whenever a probe is hit, the handler for that probe is called.

The basic syntax we used in our one-line script was to write a probe for an event, and the handler to run when that event occurred:

```
probe <event> { handler }
```

In this syntax:

- `event` is one of the `kernel.function` , `process.statement` , `timer.ms` , `begin` , `end` , or `(tapset)` aliases. For more information, look at the man page for `stapprobes` .
- `handler` can have:
  - filtering/conditionals (`if … next`)
  - control structures (`foreach` , `while`)

In the script, you don't need to declare the type of a variable; it is inferred from the context. To make our life easier, helper functions like `pid` , `execname` , `log` , etc, are predefined. Look at the language reference guide for more information. If you have installed the package, you can find it at `/usr/share/doc/systemtap-<version>/langref.pdf` .

How to run stap

The `stap` program can be invoked with multiple syntaxes:

```
stap -e '<script>' [-c <target program>]
stap script.stp [-c <target program>]
stap -l '<event*>'
```

Tapset libraries

"Open source and problem solving attitude is the way to work at Zopper" (http://opensourceforu.com/2015/08/open-source-and-problem-solving-attitude-is-the-way-to-work-at-zopper/)

DIKSHA P GUPTA , ...

(http://opensourceforu.com/2015/07/india-rides-on-meru-meru-rides-on-open-source/)

India rides on Meru, Meru rides on open source! (http://opensourceforu.com/2015/07/india-rides-on-meru-meru-rides-on-open-source/)

DIKSHA P GUPTA , ...

CONNECT WITH US

(https://w(https://t(https://www.youtube.com/channel/U(

618861 Likes | 21893 Followers | 4870 Subscribers

(http://opensourceforu.com)

3645 Comments

INTERVIEWS

We help

In the example shown earlier, after probing on the `read` system call, we printed the name of the system call, and the arguments passed via `name` and `argstr`. This was possible because in one of the `tapset` libraries, `/usr/share/systemtap/tapset/syscalls2.stp`, the following is defined:

```
probe syscall.read = kernel.function("SyS_read").call !,
                     kernel.function("sys_read").call
{
        name = "read"
        fd = $fd
        buf_uaddr = $buf
        count = $count
        argstr = sprintf("%d, %p, %d", $fd, $buf, $count)
}
```

Tapsets provide abstraction to common probe points, and define functions that you can use in your script. They (probe aliases, not probes) are not runnable themselves.

Examples

```
$ cat syscount.stp
global syscalls
probe syscall.* { syscalls[name] += 1 }
probe timer.s(10) {
        foreach(n in syscalls- limit 5)
        printf("%s = %d\n", n, syscalls[n])
        delete syscalls
}
```

Here we have taken an associative array, syscalls. An associative array is a collection of unique keys — each key in the array has a value associated with it. Here, the name of each system call would be a unique key into the array. Whenever a system call is made, we increment the value of the element in the array that corresponds to the system call name. After 10 seconds, we print the top five system calls that were made.

```
$ stap syscount.stp
read = 116
poll = 55
ppoll = 49
setitimer = 24
writev = 22
```

Let's look at another script from which we want to get the process name and PID of the process that calls the maximum system calls. We also want to exclude the SystemTap process that launches the script (`stapio`) from consideration.

```
$ cat syscount_per_process.stp
global syscalls
probe syscall.* {
        if (execname() == "stapio")
                next
        syscalls[execname(), pid()] += 1
}
probe timer.s(10) {
        foreach([procname, id] in syscalls- limit 5)
        printf("%s[%d] = %d\n", procname, id, syscalls[procname, id])
        delete syscalls
}
```

(To immediately return from a probe handler, we use the next statement.)

Running the script yields the following:

```
$ stap syscount_per_process.stp
hald-addon-stor[1074] = 30
sendmail[1157] = 14
rtkit-daemon[1387] = 8
gdm-simple-gree[1374] = 8
gnome-power-man[1370] = 7
```

We can do other interesting stuff like aggregation, getting a call graph, and even modifying a kernel variable in the running kernel. We will cover this in next month's issue.

References

- SystemTap Project (http://sourceware.org/systemtap/)
- Slides from Josh Stone's August 4, 2008 SystemTap tutorial at the LinuxWorld Conference

(http://opensourceforu.com/2016/07/organisations-reap-benefits-open-source-controlling-security-associated/)

organisations reap the benefits of open source, while controlling security and associated risks (http://opensourceforu.com/2016/07/organisations-reap-benefits-open-source-controlling-security-associated/)

JAGMEET SINGH , J...

(http://opensourceforu.com/2016/05/cio-should-have-digitisation-on-his/)

A CIO should have digitisation on his agenda (http://opensourceforu.com/2016/05/cio-should-have-digitisation-on-his/)

RAHUL CHOPRA , ...

(http://opensourceforu.com/2016/05/our-goal-is-to-give-the-world-niche-cost-effective-technology/)

Our goal is to give the world niche, cost-effective technology solutions (http://opensourceforu.com/2016/05/our-goal-is-to-give-the-world-niche-cost-effective-technology/)

NIRAJ SAHAY , MAY...

(http://sourceware.org/systemtap/wiki/LW2008SystemTapTutorial)

- Mark Wielaard's slides from Fosdem 2010 (http://sourceware.org/systemtap/wiki/HomePage?action=AttachFile&do=view&target=fosdem-stap.pdf)
- Paper titled, "SystemTap: Instrumenting the Linux Kernel for Analyzing Performance and Functional Problems" [PDF (http://www.redbooks.ibm.com/redpapers/pdfs/redp4469.pdf)]

Related Posts:

TAGS: ASYNCHRONOUS EVENTS (HTTP://OPENSOURCEFORU.COM/TAG/ASYNCHRONOUS-EVENTS/), DEBUGGERS (HTTP://OPENSOURCEFORU.COM/TAG/DEBUGGERS/), FEDORA (HTTP://OPENSOURCEFORU.COM/TAG/FEDORA/), KDB (HTTP://OPENSOURCEFORU.COM/TAG/KDB/), KERNEL (HTTP://OPENSOURCEFORU.COM/TAG/KERNEL/), KERNEL DEVELOPERS (HTTP://OPENSOURCEFORU.COM/TAG/KERNEL-DEVELOPERS/), KERNEL MODULES (HTTP://OPENSOURCEFORU.COM/TAG/KERNEL-MODULES/), LFY SEPTEMBER 2010 (HTTP://OPENSOURCEFORU.COM/TAG/LFY-SEPTEMBER-2010/), LINUX KERNEL (HTTP://OPENSOURCEFORU.COM/TAG/LINUX-KERNEL/), LOADABLE KERNEL MODULES (HTTP://OPENSOURCEFORU.COM/TAG/LOADABLE-KERNEL-MODULES/), LTRACE (HTTP://OPENSOURCEFORU.COM/TAG/LTRACE/), MEMORY REGISTERS (HTTP://OPENSOURCEFORU.COM/TAG/MEMORY-REGISTERS/), OPROFILE (HTTP://OPENSOURCEFORU.COM/TAG/OPROFILE/), PROBES (HTTP://OPENSOURCEFORU.COM/TAG/PROBES/), SAMPLING (HTTP://OPENSOURCEFORU.COM/TAG/SAMPLING/), STACK TRACE (HTTP://OPENSOURCEFORU.COM/TAG/STACK-TRACE/), SUPPORT ENGINEERS (HTTP://OPENSOURCEFORU.COM/TAG/SUPPORT-ENGINEERS/), SYSTEM CALLS (HTTP://OPENSOURCEFORU.COM/TAG/SYSTEM-CALLS/), SYSTEMS ADMINISTRATORS (HTTP://OPENSOURCEFORU.COM/TAG/SYSTEMS-ADMINISTRATORS/), SYSTEMTAP (HTTP://OPENSOURCEFORU.COM/TAG/SYSTEMTAP/)

(http://opensourceforu.com/2016/04/what-we-like-about-open-source-is-that-we-can-tweak-it-the-way-we-want/)

What we like about open source is that we can tweak it the way we want to (http://opensourceforu.com/2016/04/what-we-like-about-open-source-is-that-we-can-tweak-it-the-way-we-want/)

RAHUL CHOPRA , A...

(http://opensourceforu.com/2016/02/iot-is-not-an-it-initiative-but-a-business-function/)

IoT is not an IT Initiative, but a Business Function (http://opensourceforu.com/2016/02/iot-is-not-an-it-initiative-but-a-business-function/)

DIKSHA P GUPTA , ...

HOW-TOS

(http://opensourceforu.com/2016/08/develop-android-apps-using-mit-app-inventor/)

Develop Android Apps Using MIT App Inventor (http://opensourceforu.com/2016/08/develop-android-apps-using-mit-app-inventor/)

MEGHRAJ SINGH B...

Monitor Logs in Real-time with

Previous Article
Joy of Programming: Why C is a Middle-Level Language! (http://opensourceforu.com/2010/09/joy-of-programming-why-c-is-a-middle-level-language/)

Next Article
Playing with User-mode Linux (http://opensourceforu.com/2010/09/user-mode-linux-setup-and-debug/)

(http://opensourceforu.com/author/neependra-khare/)

Author

Neependra Khare (http://opensourceforu.com/author/neependra-khare/)

The author is an open source enthusiast with a deep interest in Linux. He is currently working as a software engineer at KQ Infotech. He also provides training on the Linux kernel and on debugging tools.

RELATED ARTICLES

VARAD GUPTA (HTTP:... MANOJ KUMAR (HTTP:... KINSHUK SUNIL (HTT...

(http://opensourceforu.com/2016/08/23262/)

Swatch (http://opensourceforu.com/2016/08/23262/)

SWAYAM PRAKASH...

(http://opensourceforu.com/2016/08/configuring-pfsense-dual-wan-failover-mode/)

Configuring pfSense in Dual WAN Failover Mode (http://opensourceforu.com/2016/08/configuring-pfsense-dual-wan-failover-mode/)

RAJESH DEODHAR ,...

(http://opensourceforu.com/2016/07/4-best-ways-perform-hassle-free-mailchimp-integration-wordpress/)

4 best ways to perform hassle-free MailChimp integration into WordPress (http://opensourceforu.com/2016/07/4-best-ways-perform-hassle-free-mailchimp-integration-wordpress/)

NOLA ARNEY , JULY...

(http://opensourceforu.com/2016/07/publishing-app-inventor-app-google-play-store/)

Publishing an App Inventor app to Google Play Store (http://opensourceforu.com/2016/07/publishing-app-

(http://opensourcefo part-1-setting-up-directory-server/)

OpenLDAP, Part 1: Setting Up the Directory Server (http://opensourcefo ru.com/2010/02/ope nldap-part-1-setting-up-directory-server/)

(http://opensourcefo for-embedded-systems-development-part-1/)

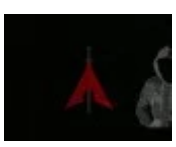
Using QEMU for Embedded Systems Development, Part 1 (http://opensourcefo ru.com/2011/06/qem u-for-embedded-systems-development-part-1/)

(http://opensourcefo letter-to-firefox-developers/)

A Letter to Foxy Developers (http://opensourcefo ru.com/2011/10/a-letter-to-firefox-developers/)

inventor-app-google-play-store/)

MEGHRAJ SINGH B...



(http://opensourceforu.com/2016/08/blackarch-linux-iso-now-comes-1500-hacking-tools/)

Insight (http://opensourceforu.com/tag/insight/) Security (http://opensourceforu.com/tag/security/) Oracle (http://opensourceforu.com/tag/oracle/) xml (http://opensourceforu.com/tag/xml/) Windows (http://opensourceforu.com/tag/windows/) FOSS (http://opensourceforu.com/tag/foss/) GNOME (http://opensourceforu.com/tag/gnome/) cloud computing (http://opensourceforu.com/tag/cloud-computing/) India (http://opensourceforu.com/tag/india/) Tips (http://opensourceforu.com/tag/tips/) web applications (http://opensourceforu.com/tag/web-applications/) html (http://opensourceforu.com/tag/html/) Applications (http://opensourceforu.com/tag/applications/) Linux (http://opensourceforu.com/tag/linux/) operating system (http://opensourceforu.com/tag/operating-system/) Networking (http://opensourceforu.com/tag/networking/) python (http://opensourceforu.com/tag/python/) Developers (http://opensourceforu.com/tag/developers/) programming (http://opensourceforu.com/tag/programming/) Microsoft (http://opensourceforu.com/tag/microsoft/) Android (http://opensourceforu.com/tag/android/) PHP (http://opensourceforu.com/tag/php/) Red Hat (http://opensourceforu.com/tag/red-hat/) open source software (http://opensourceforu.com/tag/open-source-software/) open source (http://opensourceforu.com/tag/open-source/) lets try (http://opensourceforu.com/tag/lets-try/) kernel (http://opensourceforu.com/tag/kernel/) Facebook (http://opensourceforu.com/tag/facebook/) ubuntu (http://opensourceforu.com/tag/ubuntu/) http (http://opensourceforu.com/tag/http/) database (http://opensourceforu.com/tag/database/) Java

BlackArch Linux ISO now comes with over 1,500 hacking tools (http://opensourceforu.com/2016/08/blackarch-linux-iso-now-comes-1500-hacking-tools/)

JAGMEET SINGH , AUG...

(http://opensourceforu.com/2016/08/lever-language-emerges-new-form-python/)

Lever language emerges as a new form of Python (http://opensourceforu.com/2016/08/lever-language-emerges-new-form-python/)

JAGMEET SINGH , AUG...

(http://opensourceforu.com/2016/08/facebook-open-source-fasttext-classify-bulk-text/)

Facebook releases open source FastText to let anyone classify bulk of text (http://opensourceforu.com/2016/08/facebook-open-source-fasttext-classify-bulk-text/)

JAGMEET SINGH , AUG...

(http://opensourceforu.com/tag/java/)

Apache

(http://opensourceforu.com/tag/apache/) C

(http://opensourceforu.com/tag/c/) cloud

(http://opensourceforu.com/tag/cloud/) unix

(http://opensourceforu.com/tag/unix/) LFY

April 2012 (http://opensourceforu.com/tag/lfy-

april-2012/) RAM

(http://opensourceforu.com/tag/ram/)

MySQL

(http://opensourceforu.com/tag/mysql/)

WWW

(http://opensourceforu.com/tag/www/)

firewall

(http://opensourceforu.com/tag/firewall/)

Google

(http://opensourceforu.com/tag/google/)