Getting Started with SystemTap

If programming is an art, then debugging is even more so. To be a good programmer, one must master debugging. We have seen some good methods of kernel debugging, e.g., gdb, kgdb, kprobes, etc, but none is as dynamic a tool as SystemTap, a probing and tracing tool that lets you analyse a Linux kernel's activity deeply, at runtime.

FREE NEWSLETTER

Want Daily Updates in Your Inbox?

Subscribe To Email

(https://feedburner.google.com/fb/a/mailverify?uri=LinuxForYou&loc=en_US)

CASE STUDIES

(http://opensourceforu.com/2016/03/we-have-not-faced-any-glitches-as-we-do-the-security-hardening%c2%94/)

Wehave not faced any glitches, as we do the security hardening (http://opensourceforu.com/2

SystemTap can probe system calls and kernel functions at runtime, and can examine variables in functions. There is no need to change the kernel source code to insert

instrumentation, and then recompile and install it. Simple scripts can be developed to probe the kernel at runtime.

Installation

SystemTap can be installed directly via the Synaptic package manager for Ubuntu/Debian, or Yum for CentOS, Fedora and Red Hat. However, I always like to download the source code and compile it, with the following steps.

1. Download the source code from its ftp server (http://sourceware.org/systemtap/ftp/releases/). I fetched the latest, `systemtap-1.6.tar.gz` .
2. Extract the tar ball with `tar -zxvf systemtap-1.6.tar.gz` , and change to the extracted directory (`cd systemtap-1.6`).
3. Run `./configure` . If this fails at any point (an error for a missing package) then you have to install that package (usually with `yum` / `apt-get`).
4. Once configure is done, compile it with `make` . It will take only a couple of minutes; after it's done, look for the output binary, `stap` .
5. Now run `make install` to install SystemTap. You could also add your SystemTap directory to your `PATH` variable, to directly run stap without installing.

Let's test SystemTap

This is a sample SystemTap script, which I will describe later on. First run it (as the root user, else you will get the error "Warning: /usr/local/bin/staprun is not executable (Permission denied) …").

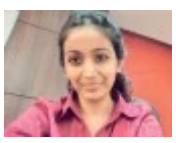
```
# cat hello.stp
probe begin
{
    printf ("hello world\n");
    exit ();
}

# stap hello.stp
hello world
```

Now, I will cover some script syntax and background details. Fundamentally, SystemTap is based on event and handler concepts. In Qt, .NET and other frameworks, this method is used for both interactive and non-interacting actions. When an event occurs, its corresponding handler will be executed. In SystemTap, when a specified event occurs, the Linux kernel runs the handler, and then resumes normal execution.

(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/)

Events are of two types, synchronous and asynchronous. Typical synchronous events are system call execution, entering and exiting a function, functions in a kernel file, etc. Asynchronous events are timers, jiffies, etc.

Handlers are written in SystemTap's script language. The format of a SystemTap script is as follows:

```
probe event { statements}
```

In a single SystemTap script, multiple events can be used. The statements to be executed for each event are enclosed in `{}` braces. When a SystemTap script is executed, the script is translated into C; the C compiler compiles the code to create a kernel loadable module (`.ko`). During the module insertion, all the events are initialised in the kernel. So when an event occurs, the corresponding handler will be run. When `exit()` is executed, the module is unloaded. To confirm this, you can omit `exit()` from the sample script and run `stap hello.stp` .

Next run `lsmod` in another terminal, to list the loaded modules. At the top of the list, you will find the `stap` module entry. Now, to exit the script, press Ctrl+C, which will unload the module.

Let us now try different events that we can use in SystemTap.

Probing a system call

```
# cat exec.stp
probe syscall.execve
{
    printf ("%s(%d) execve (%s)\n", execname(), pid(), argstr)
}
probe syscall.exit
{
    printf ("%s(%d) exit (%s)\n", execname(), pid(), argstr)
}
```

The syntax for system-call probing is `probe syscall.syscall-name` . In the above script, we probed the `execve` system call, which is used to start a new process, and `exit` , which is used to exit from a process.

Instead of a simple print statement, we have used arguments with `printf` . The function `execname()` returns the name of the current process, `pid()` the process ID, and `argstr()` the command-line arguments list. Run the script with `stap` , as follows:

```
# stap exec.stp
```

After this, I started `gnome-system-monitor` and closed it, causing the following output:

```
gnome-panel(12515) exit (0)
gnome-panel(12516) execve (/usr/lib/qt-3.3/bin/gnome-system-monitor )
gnome-panel(12516) execve (/usr/local/bin/gnome-system-monitor )
gnome-panel(12516) execve (/usr/bin/gnome-system-monitor )
gnome-system-mo(12516) exit (0)
```

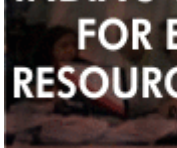
Probing a kernel function

The most important thing SystemTap can do is probe a kernel function by name. From `syscall` probing, we get only limited details; there are thousands of other kernel functions too, that we may need to probe — like network stack functions. The syntax for a kernel function probe is given below:

```
probe kernel.function("function-name") {}
```

I have chosen to probe 'ip_rcv()' as it is called very frequently, whenever an IP packet is received:

"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://(https:// (http://opensourceforu.com/)

618861 Likes | 21893 Followers | 4870 Subscribers

(http://opensourceforu.com)

3645 Comments

INTERVIEWS

We help

```
# cat ip_rcv.stp
probe kernel.function("ip_rcv")
{
    printf ("packet rcvd %s\n",$$parms);
}
probe timer.ms(4000)
{
    exit()
}
```

In the kernel function probe handler, we have used the argument `$$parms` which returns the function's parameters and its values. Other options are `$$locals` (local variables) and `$$vars` (all variables). The second handler in this script is a timer probe, whose syntax is as follows:

```
probe timer.ms(milliseconds)
```

Our handler will run after 4,000 milliseconds — thus it will `exit()` after 4 seconds. Now run the following script:

```
# stap ip_rcv.stp
packet rcvd skb=0xf50e9f00 dev=0xf412d000 pt=0xc0a4e760 orig_dev=0xf412
packet rcvd skb=0xf272ae00 dev=0xf412d000 pt=0xc0a4e760 orig_dev=0xf412
packet rcvd skb=0xf6cefa80 dev=0xf412d000 pt=0xc0a4e760 orig_dev=0xf412
packet rcvd skb=0xf436a240 dev=0xf412d000 pt=0xc0a4e760 orig_dev=0xf412
.......
```

Using jiffies

If you are a kernel programmer, you will be familiar with jiffies, a global kernel variable representing the number of ticks since the machine has booted. This can be used instead of milliseconds, in the timer. The syntax is given below:

```
probe timer.jiffies(jiffies){}
```

Here's a test script and its execution:

```
# cat variable.stp
global counter
probe timer.jiffies(100) {
    printf("count = %d\n", counter++);
}
# stap variable.stp
count = 0
count = 1
count = 2
count = 3
```

The handler will be called every 100 ticks. Like in C, we can declare global variables in the script — as we have declared 'count' and used it in the handler.

Probing kernel functions in a C file

At some time we may need to probe all functions present in a kernel C file. For example, we may want to know which functions are called during the IP input layer. The syntax is probe `kernel.function(function-name@filename)` and the test script is as follows:

```
# cat ip_input.stp
probe kernel.function("*@net/ipv4/ip_input.c")
{
    printf ("ip_input-> time=%u funcion = %s\n", gettimeofday_s(), prob
}
probe timer.ms(10000)
{
    exit()
}
```

Here we have used the file `net/ipv4/ip_input.c` and `*` instead of a function name, to invoke the handler when any function in this file is called. Now, run the following script:

```
# stap ip_input.stp
ip_input-> time=1320311892 funcion = ip_rcv
ip_input-> time=1320311892 funcion = ip_rcv_finish
ip_input-> time=1320311892 funcion = ip_local_deliver
ip_input-> time=1320311892 funcion = ip_local_deliver_finish
ip_input-> time=1320311892 funcion = ip_rcv
```

(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...

Probing function return value

To probe the return value of a function, `$$return` is used. The return value will be in a string. It can only be retrieved from a return event. The syntax is given below:

```
probe kernel.function("function-name").return {}
probe syscall.syscall-name {}
```

Test script:

```
# cat return.stp
probe syscall.mkdir.return
{
    printf ("mkdir() %s\n",$$return);
}
```

We have chosen the `mkdir()` system call (called whenever a directory is created). Execute the script, and create a directory (run `mkdir test_dir` in another terminal):

```
# stap return.stp
...
mkdir() return=0x0
```

The `mkdir` system call returns 0 on success, which has been printed by the script.

Conditional statements

In SystemTap scripts, like other programming languages, we can use if and if-else statements for branching on conditions. The syntax is simple:

```
if(condition)
    statement1
else
    statement2
```

Example:

```
# cat ifelse.stp
global counter
probe kernel.function("*@net/ipv4/ip_input.c")
{
    if (probefunc() == "ip_rcv")
    counter++;
}
probe timer.s(5)
{
    exit();
}
probe end
{
    printf ("ip_rcv() has been called %d times\n", counter);
}
```

In this script, we declared a global variable and incremented it every time `ip_rcv()` was called. After 5 seconds, the timer handler will be called and it will executive `exit()` function which in turn will exit the script. At the end, the end handler will print the counter:

```
# stap ifelse.stp
ip_rcv() has been called 12 times
```

This type of script can also be used by application developers who want to monitor kernel work when an application runs in user space.

Creating functions

Reusing source code by creating functions can also be done in SystemTap. Functions also make our scripts easy to control and read. The syntax to create a function is `function function_name(arguments) {statements}` and to use a function within a handler, it is simply `function_name(arguments)` :

(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

```
# cat func.stp
function myfunc(){
    printf("my function\n");
}
probe begin
{
    myfunc();
    printf ("hello world\n");
}
```

Let us call our simple function `myfunc()` from an event handler. The output is as expected:

```
# stap func.stp
my function
hello world
```

There are several other options in SystemTap which you can explore. For more information, refer to its official documentation (http://sourceware.org/systemtap/documentation.html).

Related Posts:

Share this:

Google (http://opensourceforu.com/2011/12/getting-started-with-systemtap/?share=google-plus-1&nb=1)

Facebook (http://opensourceforu.com/2011/12/getting-started-with-systemtap/?share=facebook&nb=1)

Twitter (http://opensourceforu.com/2011/12/getting-started-with-systemtap/?share=twitter&nb=1)

More

TAGS: C (HTTP://OPENSOURCEFORU.COM/TAG/C/), CENTOS (HTTP://OPENSOURCEFORU.COM/TAG/CENTOS/), DEBUGGER (HTTP://OPENSOURCEFORU.COM/TAG/DEBUGGER/), FEDORA (HTTP://OPENSOURCEFORU.COM/TAG/FEDORA/), KERNEL (HTTP://OPENSOURCEFORU.COM/TAG/KERNEL/), KERNEL DEBUGGER (HTTP://OPENSOURCEFORU.COM/TAG/KERNEL-DEBUGGER/), KERNEL FUNCTIONS (HTTP://OPENSOURCEFORU.COM/TAG/KERNEL-FUNCTIONS/), KERNEL PROGRAMMING (HTTP://OPENSOURCEFORU.COM/TAG/KERNEL-PROGRAMMING/), KERNEL SOURCE CODE (HTTP://OPENSOURCEFORU.COM/TAG/KERNEL-SOURCE-CODE/), LFY DECEMBER 2011 (HTTP://OPENSOURCEFORU.COM/TAG/LFY-DECEMBER-2011/), LINUX (HTTP://OPENSOURCEFORU.COM/TAG/LINUX/), LINUX KERNEL (HTTP://OPENSOURCEFORU.COM/TAG/LINUX-KERNEL/), RED HAT (HTTP://OPENSOURCEFORU.COM/TAG/RED-HAT/), SYNAPTIC (HTTP://OPENSOURCEFORU.COM/TAG/SYNAPTIC/), SYNTAX (HTTP://OPENSOURCEFORU.COM/TAG/SYNTAX/), SYSTEM CALL (HTTP://OPENSOURCEFORU.COM/TAG/SYSTEM-CALL/), SYSTEMTAP (HTTP://OPENSOURCEFORU.COM/TAG/SYSTEMTAP/), UBUNTU (HTTP://OPENSOURCEFORU.COM/TAG/UBUNTU/), YUM (HTTP://OPENSOURCEFORU.COM/TAG/YUM/)

f
(http://www.facebook.com/sharer.php?u=http://opensourceforu.com/2011/12/getting-started-with-systemtap/&t=Getting%20Started%20with%20Syste
(https://twitter.com/home?

(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-apps-google-play-store/)

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

status=Getting%20Started%20with%20SystemTap+h
started-with-systemtap/)
g+
(https://plus.google.com/share?
url=http://opensourceforu.com/2011/12/getting-
started-with-systemtap/)
p
(http://pinterest.com/pin/create/button/?
url=http://opensourceforu.com/2011/12/getting-
started-with-
systemtap/&media=http://opensourceforu.com/wp-
content/uploads/2011/12/Systemtap.jpg&description=
t
(https://www.tumblr.com/widgets/share/tool?
shareSource=legacy&canonicalUrl=&url=http%3A%2F
started-with-
systemtap%2F&posttype=link&title=Getting+Started+

inventor-in-google-play-store/)

MEGHRAJ SINGH B...

Previous Article
Device Drivers, Part 13: Data Transfer to and from USB Devices (http://opensourceforu.com/2011/12/device-drivers-transfers-to-from-usb-devices/)

Next Article
CodeIgniter: Form APIs, Validations & Pagination Techniques (http://opensourceforu.com/2011/12/codeigniter-form-apis-validations-pagination-techniques/)

(http://opensourceforu.com/author/manoj-kumar/)

Author

Manoj Kumar (http://opensourceforu.com/author/manoj-kumar/)

The author is a freelance developer and trainer. He leads a team in Linux kernel programming, Linux administration, cluster computing, embedded systems and QT/GTK programming on Linux. View and participate in the latest

discussions on his Yahoo Group (http://tech.groups.yahoo.com/group/FutureOS_Linux/).

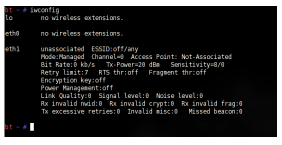
RELATED ARTICLES

SANTOSH KUMAR (HT...

(http://opensourcefo this-new-sorting-algorithm-design/)

Check This New Sorting Algorithm Design (http://opensourcefo ru.com/2015/08/che ck-this-new-sorting-algorithm-design/)

ADITYA SHEVADE (HTT...



(http://opensourcefo so-easy-to-see-your-network-activity/)

It's So Easy to See Your Network Activity, hah! (http://opensourcefo ru.com/2009/01/its-so-easy-to-see-your-network-activity/)

ANIL KUMAR PUGALIA...

The Semester Project-IV File Systems: Formatting a Pen Drive (http://opensourcefo ru.com/2013/03/the-semester-project-iv-file-systems-formatting-a-pen-drive/)

Comments Community 1 Login

Recommend Share Sort by Newest

Join the discussion…

SAS • a year ago

I am trying to probe a executable but it says: error while loading shared libraries
bit I have set my LD_LIBRARY_PATH

• Reply • Share ›

ALSO ON **OPEN SOURCE FOR YOU**

An introduction to OpenShift
1 comment • 2 months ago•
Steve Speicher — Thanks for the article, it would be great if this were updated using the

7 great apps to improve security of your Android
1 comment • 11 days ago•
Eric — Is this a paid report? Lastpass has been compromised several times.

Android vs. iOS: Which one fares well in app permission
1 comment • a month ago•
Bob Criston — Hi Ashni nice tips about app permission system, If we going through

Introduction to Selenium: an open source web automation
2 comments • 2 months ago•
Meghraj Singh — Apologies for the typo. Here are few cons of the Selenium

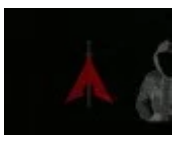
Subscribe Add Disqus to your site Add Disqus Add

TODAY WEEK MONTH

TAG CLOUD

RECENT STORIES

database (http://opensourceforu.com/tag/database/) Facebook (http://opensourceforu.com/tag/facebook/) programming (http://opensourceforu.com/tag/programming/) C (http://opensourceforu.com/tag/c/) operating system (http://opensourceforu.com/tag/operating-system/) Red Hat (http://opensourceforu.com/tag/red-hat/) lets try (http://opensourceforu.com/tag/lets-try/) unix (http://opensourceforu.com/tag/unix/) cloud computing (http://opensourceforu.com/tag/cloud-computing/) India (http://opensourceforu.com/tag/india/) html (http://opensourceforu.com/tag/html/) firewall (http://opensourceforu.com/tag/firewall/) Insight (http://opensourceforu.com/tag/insight/) Apache (http://opensourceforu.com/tag/apache/) MySQL (http://opensourceforu.com/tag/mysql/) Linux (http://opensourceforu.com/tag/linux/) operating systems (http://opensourceforu.com/tag/operating-systems/) Tips (http://opensourceforu.com/tag/tips/) Android (http://opensourceforu.com/tag/android/) xml (http://opensourceforu.com/tag/xml/) open source (http://opensourceforu.com/tag/open-source/) PHP (http://opensourceforu.com/tag/php/) WWW (http://opensourceforu.com/tag/www/) Security (http://opensourceforu.com/tag/security/) FOSS (http://opensourceforu.com/tag/foss/) Applications (http://opensourceforu.com/tag/applications/) Fedora (http://opensourceforu.com/tag/fedora/) Windows (http://opensourceforu.com/tag/windows/) Google (http://opensourceforu.com/tag/google/) http (http://opensourceforu.com/tag/http/) open source software (http://opensourceforu.com/tag/open-source-software/) Oracle (http://opensourceforu.com/tag/oracle/) LFY



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

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...

April 2012 (http://opensourceforu.com/tag/lfy-april-2012/) Developers (http://opensourceforu.com/tag/developers/)

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

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

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

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

GNOME (http://opensourceforu.com/tag/gnome/)

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

JavaScript (http://opensourceforu.com/tag/javascript/)

Networking (http://opensourceforu.com/tag/networking/)

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

web applications (http://opensourceforu.com/tag/web-applications/) Microsoft (http://opensourceforu.com/tag/microsoft/)

Creative Commons - Attribution + Noncommercial © Copyright 2016. EFY Enterprise Pvt. Ltd.