Systemtap

利用Systemtap可以很方便地对Linux系统性能和问题进行跟踪和分析。

1. 安装

安装systemtap和kernel debug info:

```
# yum install -y systemtap systemtap-runtime kernel-devel yum-utils
# debuginfo-install kernel
```

注：debuginfo-install里yum-utils包里的命令，因此需要安装。如果安装的kernel-devevl版本不对，可以从http://rpm.pbone.net/上找到相应的rpm，然后手动安装。

2. 测试

```
# stap -ve 'probe begin { log("hello world") exit () }'
```

3. 使用

曾多次在生产环境遇到重要进程被无故kill的情况，怎么查找哪个进程是始作俑者呢？在此之前我们使用的是singal-monitor中的方法，写一个kernel module，对sig处理函数进行监听。现在我们可以很方便的使用systemtap达到同样的目的了。

```
$ cat kill.stp
#! /usr/bin/env stap

probe signal.send{
  if (sig == 9 || sig == 15) {
    printf("%s %s by %s: %s(%d) -> %s(%d)\n", ctime(gettimeofday_s()),
sig_name, name, execname(), pid(), pid_name, sig_pid);
  }
}
```

输出类似：`Thu May 21 01:26:44 2020 SIGTERM by signal_generate: zsh(11513) -> sleep(11561)`。

如果要指定被kill的进程名，可以在`if`语句中增加`pid_name == "YOUR_PROCESS_NAME"`，变为`if ((sig == 9 || sig == 15) || pid_name == "YOUR_PROCESS_NAME")`。

4. 生成内核模块

正常情况下，stap会动态生成内核模块并加载。如果所在的机器加载内核时需要签名，我们可以静态生成内核模块，签名后再运行该模块。

```
# stap kill.stp -p 4 -m catch_kill.ko
# staprun catch_kill.ko # 运行该内核模块
```

5. 参考资料

- Systemtap tutorial: https://sourceware.org/systemtap/tutorial.pdf
- Get Involved systemtap: https://sourceware.org/systemtap/getinvolved.html
- Using systemtap: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/systemtap_beginners_guide/using-systemtap
- 用 SystemTap 找出送 SIGKILL 的 process: https://medium.com/fcamels-notes/%E7%94%A8-systemtap-%E6%89%BE%E5%87%BA%E9%80%81-sigkill-%E7%9A%84-process-2f0034aa92e0
- SYSTEMTAP: KILL() [WHO KILLED MY PROCESS?]: https://www.ibm.com/developerworks/community/blogs/IMSupport/entry/SYSTEMTAP_KILL_WHO_KILLED_MY_PROCESS?lang=en
- Tapset syskill: https://sourceware.org/systemtap/tapsets/API-signal-syskill.html
- probe::signal.send: https://sourceware.org/systemtap/tapsets/API-signal-send.html