SystemTap

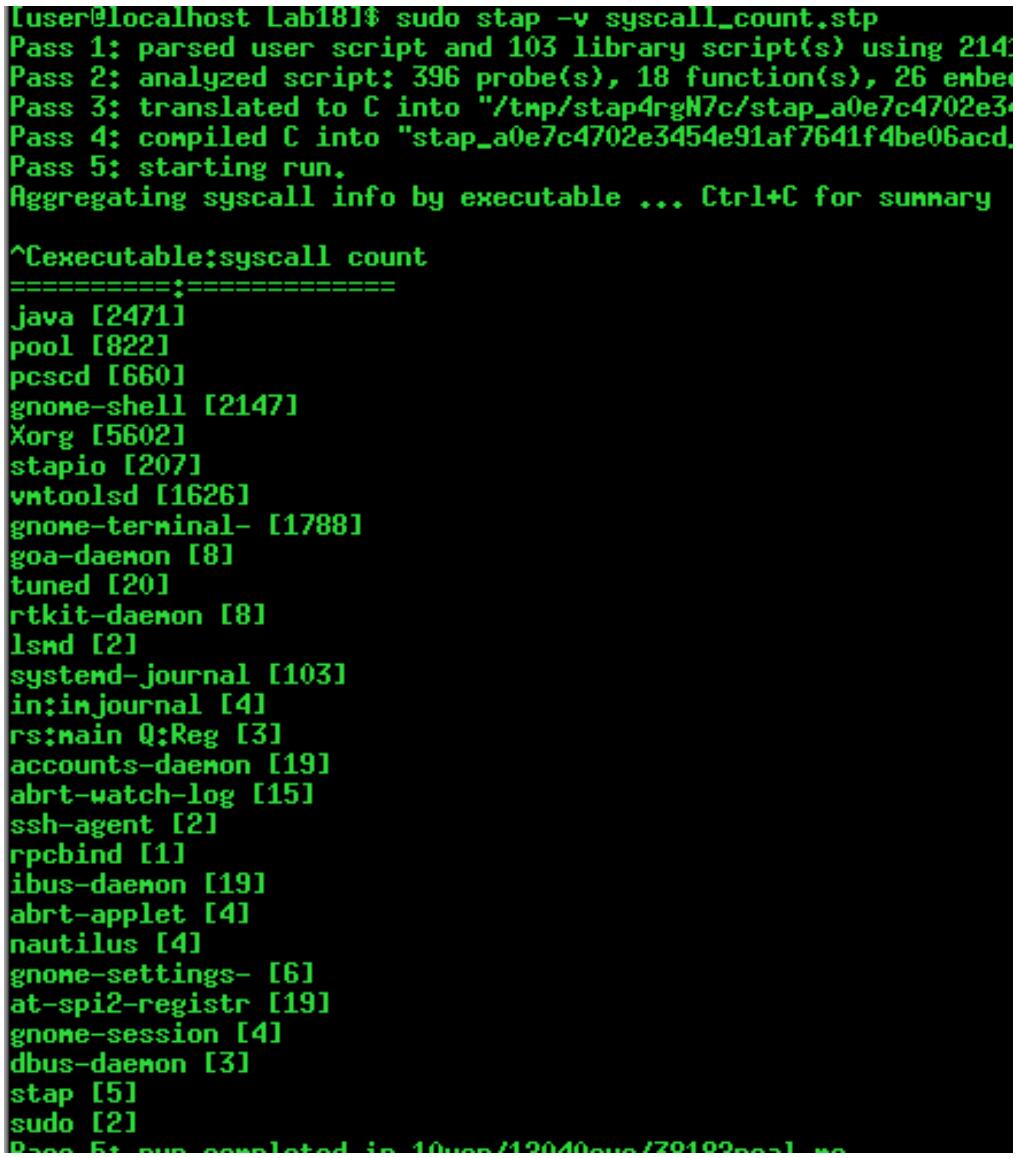
Objective: In this lab you will develop a SystemTap script that aggregates syscall counts by executable name.

Note: Make sure you are not running the debug version of the kernel (uname –a) since the SystemTap will not work for that one.

File(s) for this lab:

1. Complete a SystemTap script that probes all syscalls to the kernel, and keeps count of the total number by the name of the executable that initiated the call.

2. The output should look like the following:


[user@localhost Lab18]$ sudo stap -v syscall_count.stp
Pass 1: parsed user script and 103 library script(s) using 214
Pass 2: analyzed script: 396 probe(s), 18 function(s), 26 embe
Pass 3: translated to C into "/tmp/stap4rgN7c/stap_a0e7c4702e3
Pass 4: compiled C into "stap_a0e7c4702e3454e91af7641f4be06acd.
Pass 5: starting run.
Aggregating syscall info by executable ... Ctrl+C for summary
^Cexecutable:syscall count
==========:==========
java [2471]
pool [822]
pcscd [660]
gnome-shell [2147]
Xorg [5602]
stapio [207]
vmtoolsd [1626]
gnome-terminal- [1788]
goa-daemon [8]
tuned [20]
rtkit-daemon [8]
lsmd [2]
systemd-journal [103]
in:imjournal [4]
rs:main Q:Reg [3]
accounts-daemon [19]
abrt-watch-log [15]
ssh-agent [2]
rpcbind [1]
ibus-daemon [19]
abrt-applet [4]
nautilus [4]
gnome-settings- [6]
at-spi2-registr [19]
gnome-session [4]
dbus-daemon [3]
stap [5]
sudo [2]