LAB 02
DUE – 1 week by 12:15 PM to D2L

CSCI 4417 System Administration

Process Management

Purpose

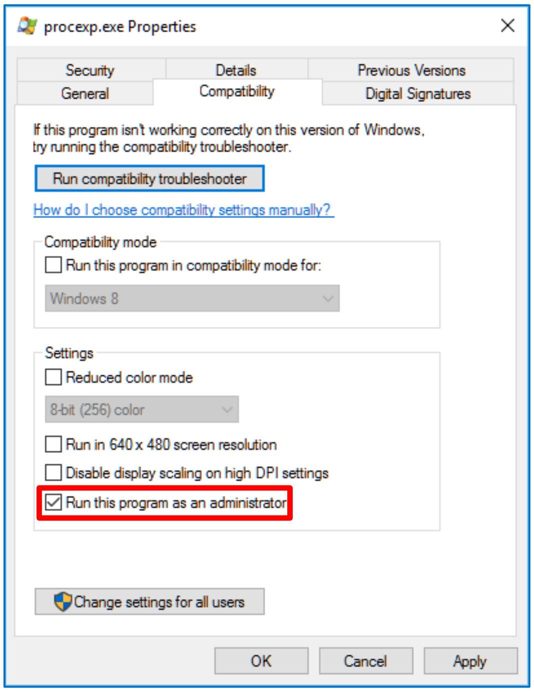
Explore some of the process management tools available for Windows and Ubuntu

Required

1) SysInternals Process Explorer

Windows Process Management

1) If you haven't already, download and extract the files for the SysInternals Suite.
2) I usually keep Process Explorer on the Desktop, but you can store it wherever you want. It's a standalone application, so there's no installation
3) While we could copy Process Explorer to one of our Windows servers, it will probably be a lot more interesting to run it on a local machine – there'll be more processes running, since we haven't install any Windows services on our instances yet
4) If you are using a lab computer, you can run Process Explorer from an external drive
5) You will have more success if you ***right click, run as Administrator***
    a. You can make it run as administrator by default by right-clicking the icon and selecting 'Properties.'
    b. Click on the 'Compatibility' tab
    c. Check the 'Run this program as an administrator and 'OK'



procexp.exe Properties
Security
Details
Previous Versions
General
Compatibility
Digital Signatures
Run compatibility troubleshooter
How do I choose compatibility settings manually?
Change settings for all users
OK
Cancel
Apply

If this program isn't working correctly on this version of Windows, try running the compatibility troubleshooter.

Compatibility mode

☐ Run this program in compatibility mode for:

Windows 8

Settings

☐ Reduced color mode

8-bit (256) color

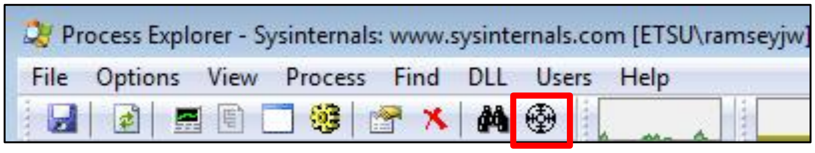
☐ Run in 640 x 480 screen resolution

☐ Disable display scaling on high DPI settings

☑ Run this program as an administrator

Figure 1: Setting up Process Explorer

6) Launch Process Explorer (as Administrator)
7) If the bottom pane isn't showing, type 'Ctrl-d'
8) Investigate and write up the following about Process Explorer
    a. What is the logic for the grouping of the Processes? (Remember? From lecture)
9) Open up notepad.exe then in Processes Explorer double click the entry relating to notepad. Why is it located under explorer.exe? Why do you think 'Autostart Location' is 'n/a'? Take a moment to examine each tab. You may not be able to view Threads. If not, don't worry about it
    a. Examine Notepads DLLs. What is gdi32.dll? Hint: if you don't already know, right-click on it and see if any of the options that appear might clear things up for you
10) Right-click on the notepad.exe process
11) Select 'Suspend'
12) Now, try clicking on the Notepad window
13) This time, right-click on notepad.exe and select 'Kill Process'
14) What does this do? (See image below). What happens? How could this feature be of use to an administrator?



Process Explorer - Sysinternals: www.sysinternals.com [ETSU\ramseyjw]
File
Options
View
Process
Find
DLL
Users
Help

Figure 2: Process Explorer tools

15) Launch cmd.exe

16) In the console window, enter the following command `ping -t google.com`. The “-t” switch causes Windows ping to ping continuously until stopped. We’ll use this to simulate a runaway process.

17) What happens if we attempt to kill the process – right-click on cmd.exe and select ‘Kill Process’ -- (keep an eye on cmd.exe’s tree when you kill the process. What happens? How do we make it stop pinging?)

18) Find ping in the tree. It is now running as a direct child of explorer.exe. Ewwww! Kill it!

19) Now, repeat Steps 15 and 16

20) This time, right-click on cmd.exe and select ‘Kill Process Tree’. *That* took care of it!

21) You can refresh your memory with regard to what the colors mean by clicking on ‘Options’ and selecting ‘Configure Colors’

22) There is another feature available in Process Explorer that makes it extremely useful in chasing down malware

23) Click on ‘Options’

24) Hover over VirusTotal.com and click on ‘Check VirusTotal.com’

25) You’ll have to agree to the TOS (a no-brainer)

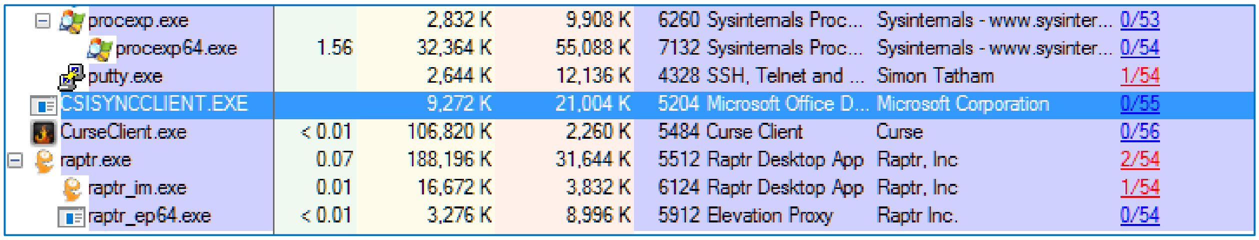
26) What happens next is that Process Explorer will check *every running process* against VirusTotal – 50+ separate virus scanners

27) There are a couple of catches:

    a. It only checks running processes, so if you have malware on your machine that isn’t active, it won’t find it. But if it isn’t active, it isn’t hurting you

    b. It won’t detect rootkits. You need specialized software for that

    c. Sometimes one or two of the scanners will report back that the software is suspicious. Most often, these are false-positives. Take a look at this screen shot from my computer:



procexp.exe		2,832 K	9,908 K	6260 Sysinternals Proc...	Sysinternals - www.sysinter...	0/53
procexp64.exe	1.56	32,364 K	55,088 K	7132 Sysinternals Proc...	Sysinternals - www.sysinter...	0/54
putty.exe		2,644 K	12,136 K	4328 SSH, Telnet and ...	Simon Tatham	1/54
CSISYNCCLIENT.EXE		9,272 K	21,004 K	5204 Microsoft Office D...	Microsoft Corporation	0/55
CurseClient.exe	< 0.01	106,820 K	2,260 K	5484 Curse Client	Curse	0/56
raptr.exe	0.07	188,196 K	31,644 K	5512 Raptr Desktop App	Raptr, Inc	2/54
raptr_im.exe	0.01	16,672 K	3,832 K	6124 Raptr Desktop App	Raptr, Inc	1/54
raptr_ep64.exe	< 0.01	3,276 K	8,996 K	5912 Elevation Proxy	Raptr Inc.	0/54

Figure 3: Screen shot from Process Explorer with VirusTotal activated

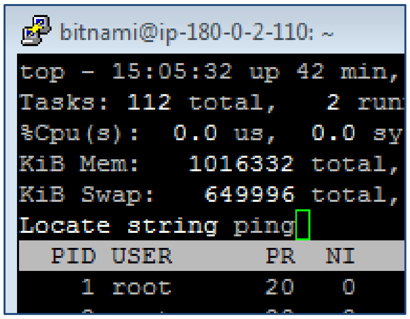
28) Note that one scan thought *PuTTY* is suspicious (last column)!

29) Finally, remember that if you’re working from your personal computer and are impressed with Process Explorer over Task Manager, you can swap it with Task Manager under the ‘Options’ menu

Linux Process Management

`top`

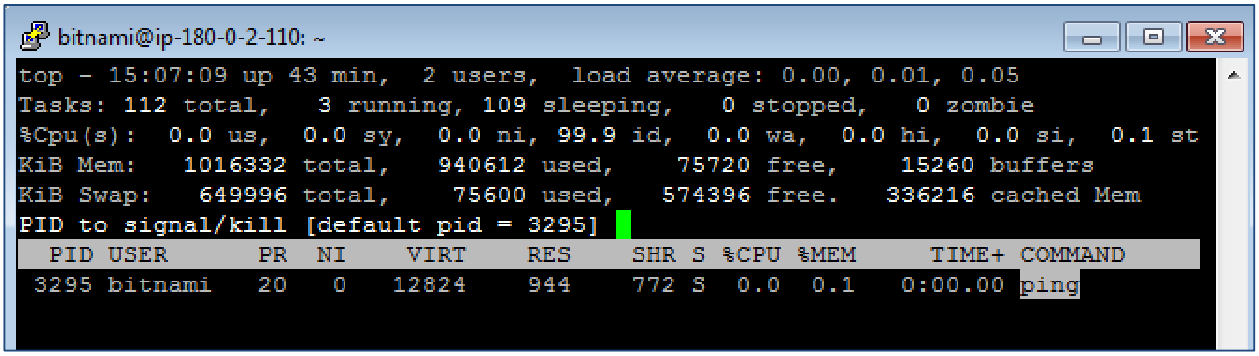
1) Launch your Ubuntu instance and log in
2) In your Ubuntu instance, issue the **`top`** command
3) Define every portion of the output for the **`top`** command (hint: try man top and look in section, maybe, 3), i.e., “PID,” “USER,” “PR,” “NI,” etc. Type ‘q’ to exit **`top`** to do so, or open a second PuTTY window (Step 8) now
4) If you exited out of **`top`** for Step 3, run it again
5) Type the ‘h’ key to bring up **`top`**’s help. Take a look at some of the switches
6) Type ‘q’ to exit help (you can refer back to it any time)
7) Type ‘Shift-v’ for a forest view, showing dependencies (much like, though not as pretty as, Process Explorer). Type ‘Shift-v’ to toggle the forest view off
8) Open another terminal window by launching another instance of PuTTY and logging in to your instance on it. In this window, run **`ping google.com`**. This, again, will simulate our rogue process that must be terminated.
9) Return to the **`top`** terminal display.
10) Type ‘L’ (i.e., Shift-L) and enter ‘ping’ as the search string.



```
top - 15:05:32 up 42 min,
Tasks: 112 total,   2 run
%Cpu(s):  0.0 us,  0.0 sy
KiB Mem:   1016332 total,
KiB Swap:   649996 total,
Locate string ping
  PID USER       PR  NI
    1 root       20   0
```

Figure 4: Searching for a running process

11) type ‘k’ (lowercase ‘k’)
12) **`top`** will ask which PID to kill. Since you did the search, the PID should already be selected. Press ‘Enter’. Note that the default signal is SIGTERM, which, as we discussed in class, is the ‘nicer’ way to terminate a process.



```
top - 15:07:09 up 43 min,  2 users,  load average: 0.00, 0.01, 0.05
Tasks: 112 total,   3 running, 109 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.1 st
KiB Mem:   1016332 total,   940612 used,    75720 free,    15260 buffers
KiB Swap:   649996 total,    75600 used,   574396 free.   336216 cached Mem
PID to signal/kill [default pid = 3295]
  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 3295 bitnami   20   0   12824    944    772 S  0.0  0.1   0:00.00 ping
```

Figure 5: Killing a process with top

13) Observe the terminal window ping was running in. What happened?

ps aux (with a little **less** and **grep**)

14) While top is still running, issue **ps aux | less** command in the second terminal window. Piping the output of **ps aux** through **less** allows us to examine the output one page (or line) at a time (space bar advances a page, the down arrow a line at a time)
15) Page down to the end of the file. Note that the line for **top** has a TTY entry (seventh column) of ‘pts/0’ while **ps aux** is pts/1. What could that possibly mean?
16) Return to the console window by typing ‘q’ to quit less
17) Enter: **who**. Does that clear things up regarding TTYs?
18) Let’s say our Apache server is doing something squirrelly. Enter

```
ps aux | grep “www-data”
```

Note: On an Ubuntu machine, www-data is the user/group that owns apache2.

19) How might the output from this command assist in troubleshooting?
20) While top is still running, in a second terminal window, kill the top command using

```
sudo kill -15 [pid for top]
```

Look what happened in the first window.

21) Why might a static dump of currently running processes be preferable to top’s more dynamic display? (by the way, using the command line switches -b (batch) and -n1, i.e., top -b -n1) Think: scripting and automation…

htop (with a little **apt-get install**)

22) Let’s take a quick look at **htop**. First we have to install it.
23) Enter **sudo apt-get install htop -y**
24) When you’re returned to the command prompt, enter **htop**.
25) Note that the same information is displayed that top displays, but the interface is somewhat more friendly.

26) If you wish, run the ping command again in the second terminal window and use `htop` to kill it. (after you search for ping, enter F9. The ping line will still be highlighted, as will the SIGTERM signal. Pressing Enter will then terminate `ping`).

/proc

27) `proc` is a pseudo-directory that also shows every running process. So if we long-list (`ll`) `/proc`, we'll get a listing of processes in another way
28) Enter `ll /proc | less` and observe the output. Press 'q' to exit `less` when you're done
29) Now, enter `ll /proc | grep "ubuntu"` to see the processes you own
30) proc also contains a lot of system information in 'subdirectories'
31) Enter `cat /proc/cpuinfo` (remember from 2200 – 'cat' will dump the contents of a text file to STDOUT (the display))
32) When you're done, enter `sudo shutdown now`, return to the AWS EC2 console, and shut it down from there as well

---

Complete your lab report in accordance with the Lab Guide. Answer the questions included in this lab as explanations following the steps you list. Your hypothetical audience is some future junior sysadmin who's using your documentation as a reference. Perhaps, he or she needs to examine his/her Ubuntu instance (or VM) because of lagging performance, for example.

A final note: Conforming to the instructions for lab reporting in the Procedure section is going to be a bit awkward for this lab, as the instructions are provided to you and are quite detailed (detailed enough, I hope). However, in future labs, I hope to assign a task or series of tasks, for which you will have to formulate your own solution. I think it'll make more sense and provide greater value then. Remember also, and I meant to bring this up in class but forgot: The tone of your Procedure section should be professional and written in third person. A lot of the first reports I graded used informal language. Words like 'you,' 'I,' 'we,' 'me,' 'your,' etc., should not appear in the Procedure section. When you're recording your Observations, this isn't so much of an issue. Wherever you find employment in the future, there will be guidelines and standards for documentation, and "This is the way I've always done it" won't cut the mustard.

⚠ ***Please remember: When you are done working in AWS, shut down all your running instances before logging out!***