Notes from Exploring Programs

Instructions

Make a copy of this document, rename it to “exploring-programs-notes” and move it to your CSE 523 Google Docs collection. If at any point in this exercise you feel stuck, raise your hand and get some guidance. When you reach each GATE below, switch over to the Tracking Progress document and update your position. Try to be efficient with your time.

Overview

Today we will explore program binaries and running processes. Keep detailed notes below (place your comments in between the provided horizontal lines); you will be referring to these in the future to do your work.

Part 1: Linux

For this part, you will be working in your Ubuntu VM, so start that now.

1.1 Static Binaries

In this exercise, we will be working with the Linux utility **date**. Enter the following command lines in a terminal window.


```
date
date --date=”4 hours ago”
date --date=”2 years ago” +%Y%m%d
```

Where is the program **date** in the filesystem? Show how you found it below (ie, show the command line and its output.)

PLACE YOUR TEXT HERE BETWEEN THESE SEPARATORS

Briefly describe what the program **objdump** does.

Enter the following command at the terminal, replacing {loc} with the location you found above (you probably want to find-replace {loc} for your location so that you can copy and paste

directly). This will pipe the output of `objdump -d` into a program called `less`. Here are some quick and easy commands to learn to make `less` easier to navigate. Note that, much like `Vi(m)`, `less` interprets your keypresses as commands at the output window.

Scroll up (J), scroll down (K), start a forwards search for a string (/), start a backwards search for a string (?), quit less (q), scroll down one window (SPACE), scroll forwards ½ window (d), and scroll back ½ window (b). Now you know enough to be quick using `less`!

Explain what you see below, using only a few sentences.

objdump -d {loc}/date | less

GATE 1

Enter the following command at the terminal, contrast this output with the previous output, and explain what the -xtrds switches mean.

objdump -xtrds {loc}/date | less

Enter the following command at the terminal, and note the output below.

objdump -xtrds {loc}/date | grep bugs

GATE 2

Make a copy of **date** in your home directory. Launch **ghex** . Briefly describe the tool and its purpose below.

Open your local copy of date in **ghex**. Look familiar? Go to Edit > Find, type in “bugs” into the right pane and click “Find Next”. In the right-hand column, you should see the string “bugs” on

the last line. At the cursor's present location on the left side, change the hex numbers "62 75" to read "64 6F". Save and quit with File > Save and File > Exit. Go back to the terminal and run (the local version of) "./date --help". Explain the differences below.

Use ghex to make another, more interesting change to your local copy of date, and explain it below.

GATE 3

Another useful tool is available: readelf. ELF is the object file format used in Linux and many other systems. Execute the following commands at the terminal, and explain what each one shows you. If you'd like, you can keep a copy of each output in this file for future reference.

readelf -l {loc}/date
readelf -S {loc}/date
readelf -W -s {loc}/date
readelf -x 16 {loc}/date

GATE 4

1.2 Using gdb

At the terminal, enter the following
gdb
(gdb) help

Use the interactive help feature to explore gdb's options.

GATE 5

1.3 Examining processes

At the terminal, enter the following.
gdb date

(gdb) run
(gdb) quit
Copy your output below.

Try the following commands to get a feel for the tool.
gdb date
(gdb) set args “--help”
(gdb) break __libc_start_main
(gdb) run
(gdb) frame
(gdb) bt
(gdb) info frame
(gdb) info registers
(gdb) x /16xw $rsp

Explore other registers and memory locations. What can you say about where the code and stack are located?

GATE 6

Now try the following.

gdb date
(gdb) set args “--version”
(gdb) break __libc_start_main
(gdb) run
(gdb) maintenance info sections
note the start address of the .rodata section
(gdb) x/20s {.rodata address}
note the start address of the string “GNU coreutils”, {start}
(gdb) set *{start} = 0x20554c42
(gdb) c

What is the code above doing? Can you modify it to do something else that demonstrates that you understand? Did you run into a snag? Provide your explanation and your modification below. A reference like this might be helpful: http://www.asciitable.com

COMPLETE