Classwork Assignment

Name – Ashutosh Soni

Id- 2018ucp1505

Question 1: Can buffer be still utilized for stack overflow if both buffer and stack grow in same direction (say low memory to high memory)? If yes, explain how

Answer: The direction of stack growth has nothing to do with whether or not it overflows. At the very least, you need to explain why you think otherwise. Stacks overflow because the allocated segments for the stack are exhausted, and that's a notion independent of direction. Creating a runtime whose stack doesn't overflow requires (a) recognizing overflow and (b) doing something to allocate new frames. Typically, the various implementations of functional languages use a variety of techniques to enable very deep stacks.

Question 2: How can NX (or its equivalent) bit be set or reset on Intel architecture?

Answer: Firstly Restart your computer, press F2,F12 or DEL to enter the BIOS Setup menu and then go to security lab settings, find the option called "NX Bit" , "Execute Disable bit" or "XD bit" and then restart your PC.

Question 3: How can ASLR be enabled (or disabled) in Linux based systems?

Answer: We can configure ASLR in Linux using the /proc/sys/kernel/randomize_va_space interface.

The following values are supported:

0 – No randomization. Everything is static.

1 – Conservative randomization. Shared libraries, stack, mmap(), VDSO and heap are randomized.

2 – Full randomization. In addition to elements listed in the previous point, memory managed through brk() is also randomized.

So, to disable it, run

echo 0 | sudo tee /proc/sys/kernel/randomize_va_space

and to enable it again, run

echo 2 | sudo tee /proc/sys/kernel/randomize_va_space

> This won't survive a reboot, so you'll have to configure this in `sysctl`. Add a file `/etc/sysctl.d/01-disable-aslr.conf` containing:
>
> ```
> kernel.randomize_va_space = 0
> ```
>
> should permanently disable this.

Question 4: How can memory map of a process be obtained on your machine? What do you infer from this map?

Answer: The generic way is /proc/iomem. That shows you the kernels of view of what memory ranges are assigned to who.

If you want more detail you'll need to look at each individual driver.

You might get some more information from /proc/vmallocinfo because ioremap() uses vmalloc (though possibly not on all architectures).

The other way is The pmap command in Linux is used to display the memory map of a process. A memory map indicates how memory is spread out.

Syntax:

pmap [options] pid [...]

Question 5: Disable ASLR and NX on your system. Write a simple C program in which atleast 3 activation frames are pushed to stack at a given time? Using a suitable code with pointer, explore the structure to identify in which order parameters, frame pointer, return address and local variables are pushed? In other words, what is structure of activation frame in your machine?

Answer: To disable ASLR, you need to put a 0 into /proc/sys/kernel/randomize_va_space. You can go with

```
sudo -i
```

which will bring a root console, and then

```
echo 0 > /proc/sys/kernel/randomize_va_space
```

This will disable ASLR at kernel level.

For the NX bit, you need to compile your program with the -fno-stack-protector -z execstack GCC flags:

```
gcc -fno-stack-protector -z execstack -o your_executable your_source.c
```

This will require you to install execstack tool that removes the NX bit from executables