Assignment 3

Hyoungshick Kim

December 4, 2020

1 Writing an exploit

This is a programming assignment to understand memory corruption attacks and how to exploit a program. You will carry out the project using a virtual machine, on your own computer.

- This project will use a virtual machine in the VirtualBox (`https://www.virtualbox.org/`) format. The overall VirtualBox manual is available here (`https://www.virtualbox.org/manual/UserManual.html`).
- The first step is to install VirtualBox on your computer; our VM images have been tested with version 5.2.2. There are specific instructions for installing VirtualBox for computers running Windows (`https://www.virtualbox.org/manual/ch02.html#installation_windows`), Linux (`https://www.virtualbox.org/manual/ch02.html#install-linux-host`), and Mac OSX (`https://www.virtualbox.org/manual/ch02.html#idp52693904`).
- The next step is to download the virtual machine image, in OVF format, that we will use for the project. It has extension .ova meaning it is an archive with all of the relevant materials in it. The file can be found here (`https://drive.google.com/file/d/1MTNadwQGWHDiU8kHINoquFyDzfR71I8H/view?usp=sharing`). The virtual machine runs a version of Ubuntu Linux.
- Finally, you must import this OVF file and run it. To import it, it should be as simple as double-clicking the .ova file. Doing so will start VirtualBox and ask you whether to import it the image. You should then click "import". Further instructions are available here (`https://www.virtualbox.org/manual/ch01.html#ovf`).
- Having imported the VM, you should see it in your list of VMs. Select it and click "Start". This will open a window running the virtual machine, starting up Ubuntu Linux. When you get to a login screen, use username **normaluser** and password is **P@ssw0rd**. Then start up a terminal window.
- We have placed a compiled program `main` in the home directory in the virtual machine.
- The program reads data from a file and writes the data to stdout (i.e., the terminal). You can run the program by typing ./main [file name] on the command prompt.

- This program is vulnerable to a memory corruption attack. Your job is to implement an exploit (`attack.c`) producing `input.txt` for this vulnerable program (`main`) in order to run an interactive shell as *root*. You can use `id` command to check if it's being run as *root*. To demonstrate your attack, we run the following commands:

```
gcc -o attack attack.c
./attack
./main input.txt
```

  - To exploit the program, you may need to use a shellcode. A possible example shellcode (`shellcode.nasm`) is as follows:

```
xor     eax, eax     ;Clearing eax register
push    eax          ;Pushing NULL bytes
push    0x68732f2f   ;Pushing //sh
push    0x6e69622f   ;Pushing /bin
mov     ebx, esp     ;ebx now has address of /bin//sh
push    eax          ;Pushing NULL byte
mov     edx, esp     ;edx now has address of NULL byte
push    ebx          ;Pushing address of /bin//sh
mov     ecx, esp     ;ecx now has address of address
                     ;of /bin//sh byte
mov     al, 11       ;syscall number of execve is 11
int     0x80         ;Make the system call
```

  - To compile it use `nasm`:

```
nasm -f elf shellcode.asm
```

  - Use objdump to get the shellcode bytes:

```
objdump -d -M intel shellcode.o
```

- If you write another exploit code to successfully launch an attack when we use the address randomization, you will get an extra credit. You can enable and disable ASLR using `enable_aslr` and `disable_aslr` commands, respectively.

- You also need to write a report (in a separate file) to explain your code in detail. **Your assignments must be your own original work.** We will use a tool to check for plagiarism in assignments.

- Please upload your source code (`attack.c`) for the exploit and instructions to illustrate how your source code works to iCampus.