Attacker Machine

As I said earlier, the attacker machine will be our main base and I prefer Kali Linux, but if you

are going to use another distribution, then you have to install the following packages:

* First, we need to make sure that the C compiler is installed; use the gcc -v command:
* If not, just install it using $ sudo apt-get install gcc (Debian distributions) or $ sudo yuminstall gcc (Red Hat distributions). Accept and install gcc with its dependencies.
* Also, we are going to use the Python programming language in exploit development.Python comes by default with most Linux distributions, and to make sure that it's installed,just use $ python -V or just python. Then, the Python interpreter will start (hit *Ctrl* + *D* to exit):
* For text editors, I use nano as my CLI text editor and atom as my GUI text editor; nano alsocomes with most Linux distributions.
* As we are going to use assembly programming language, let's take a look at the assembler (nasm) and the linker (ld).
* First, we need to install nasm by using $ sudo apt-get install nasm (Debiandistributions). For Red Hat distributions, according to NASM's website, you first need toadd this repository to your /etc/yum/yum.repos.d as nasm.repo:
* Then, use $ sudo yum update && sudo yum install nasm to update and install nasm and $ nasm -v to get NASM's version:

Victim Machine

This machine will be Ubuntu 14.04 x64. You can download it

from http://releases.ubuntu.com/14.04/. Also, we have to follow previous instructions for gcc, Python, and nasm.

**$ sudo apt-get install cmake build-essential libboost-dev libqt5xmlpatterns5-dev qtbase5-dev qt5-**

**$ git clone --depth=50 --branch=3.0.4 https://github.com/aquynh/capstone.git**

**$ pushd capstonep**

**$ ./make.sh**

**$ sudo ./make.sh install**

**$ popd**

**$ git clone --recursive https://github.com/eteran/edb-debugger.git**

**$ cd edb-debugger**

**$ mkdir build**

**$ cd build**

**$ cmake ..**

**$ make**

Then, start edb-debugger using the $ sudo ./edb command, which opens the following window:



Now to the final step. It's necessary to disable **Address Space Layout Randomization** (**ASLR**)

for learning purposes. It's a security mechanism in Linux, and we will talk about it later.

Just execute the $ echo 0 | sudo tee /proc/sys/kernel/randomize\_va\_space command.

Also, we are going to disable the stack protector and NX when using gcc when compiling is

done, using:

**$ gcc -fno-stack-protector -z execstack**

**Windows Victim Machine**

Here, we are going to configure a Windows machine as a victim machine, which is Windows 7

x64.

First, we need to install C compiler and IDE, I suggest *Code::Blocks*, and to install it, download

the binary from http://www.codeblocks.org/downloads/binaries. Here, I'm going to install codeblocks-

16.01mingw-setup.exe (the latest version). Download and install the mingw version.

At the first boot of *Code::Blocks*, a window will pop up to configure the compiler. Select GNU

GCC Compiler, click Set as default, and then click OK:

First, we need *Immunity Debugger* for x86; download Immunity from https://debugger.immunityinc.c

om/ID\_register.py. Fill this form in, download, and then install it using the default settings, and it

will ask you to confirm installing Python. After that we need to install a plugin for a debugger

named mona, created by the Corelan team, https://www.corelan.be. It's a wonderful plugin that will

help us in exploit development. Download the mona.py file from their GitHub repository, https://git

hub.com/corelan/mona, then copy it to C:\Program Files (x86)\Immunity Inc\Immunity

Debugger\Immunit\PyCommands:

Go to https://sourceforge.net/projects/x64dbg/files/snapshots/, then download the latest version.