CS3235 - Assignment 2 - Memory Corruption Attacks

Each exercise's folder contains a script that runs the exploit as described below. To use it simply run:

```
chmod +x ./exploit.sh
./exploit.sh
```

The required payloads are generated and written into their files using python scripts. For buffer overflow, the executable is ran once to collect the address of the buffer that is then used to build the payload. Even though ASLR should be off this was first implemented as the addresses may change when the system reboots.

These exploits were designed for a 64 bit Ubuntu 16.04 LTS machine. For details on how the payloads are generated consult the source code of the Python scripts included.

1 - Buffer Overflow

The script build_exploits.py automatically generates the files (exploit1 and exploit2) such that running `./buffer_overflow` causes a shell to be spawned.

To run the attack just use the sequence of commands:

```
make
python3 build_exploits.py
./buffer_overflow
```

2 - Format String Attack

The script build_exploit.py automatically generates the file (exploit) such that running `./format_string < exploit` causes the program to print "You won!"

To run the attack just use the sequence of commands:

```
make
python3 build_exploit.py
./format_string < exploit
```

3 - Return-Oriented Programming

The script build_exploit.py automatically generates the file (exploit) such that running `echo -ne 'run\n-1' | gdb rop` causes the chosen file to be printed to stdout.

This exploit does NOT work outside of gdb.

To run the attack just use the sequence of commands, replacing <FILENAME> by the name of the file to leak. Note that it be at most 55 characters long.

```
make
python3 build_exploit.py <FILENAME>
echo -ne 'run\n-1' | gdb rop
```

Limitations

Please note that this exploit can only print complete files if their size is up to 24576 bytes. For larger files, one could use the buffer iteratively and print one chunk of up to 24576 bytes at a time. For the purpose of this exercise it seems unnecessary to implement such solution.