**Exploit 1**

Target 1 had a buffer overflow vulnerability.

The size of buf[] was 512 bytes. Buf[] copied the command line argument argv[1] directly into itself. Hence, this struck me to use buffer overflow exploit. Additionally, 17 bytes were required to exploit the buf[]. So, in total I used a buffer of size 529 i.e 512+16+1 bytes. The additional 1 byte is for null or terminating character. This covered the distance from the start of the target buffer(buf[]) to return address. For this exploit shellcode provided in the archive was used.

**Exploit 2**

Target 2 had a buffer overflow and an integer overflow vulnerability.

First I got past the integer value check in function foo and then exploited buffer flow. The integer overflow is possible in the instruction count \* sizeof(struct widget\_t). A value larger than 2147483647 would roll over, so I set the value to be -2147483647. I got a value of 28020. I filled a buffer, starting at 0, with value -2147483647, followed by enough NOP instruction for the rest of the buffer size of 28028. The last four bytes of the buffer was used for the address of buf in the target program and the area immediately before the buff address was used for the shellcode that was given to us to use in this project.

**Exploit 3**

Target 3 has heap overflow vulnerability.

It has verbose description in the slides. The programmer misused tfree. I exploited this vulnerability to insert a chunk that held the shellcode. For this, the address of the chunk on the left was required. Once found I overwrote the address it pointed to the chunk I had developed. The code for this exploit was given in the slides so I was able to modify the addresses along with the size of the buff constructed in the exploit to expose the shell.

**Exploit 4**

Target 4 has format string vulnerability.

As the printf directly passes the arguments it is highly vulnerable to format string attacks. The dtors section is a destructor and is writable section and is always executed after the main function get executed. Hence, what I did was I found out the address of dtor end and overwrote the shellcode in that part leading the target to get exploited. In this way I got the root shell privilege.