1. To find address of any method, from gdb, use

*print <method\_name> //*

1. To get location of any string in memory location of a program, follow below steps
   1. (gdb)*info proc map*  // this will return start and end address of all libraries that a given binary is using.
   2. (gdb)*find <start\_address>,<end\_address>,<string within double quotes>. //* this will find any occurrence of given string within specified memory range.
2. To check if stack is executable OR not

*readelf -l <binary\_name> | grep -i -A1 stack* // if stack has RWE OR RWX set, it means stack is executable.

1. To check for endianness, use

(gdb)*show endian*

1. To look for any methods in shared object (so) files, use
   1. *readelf -s <shared\_object\_path> | grep <string>*
   2. objdump -D *<shared\_object\_path> | grep <string>* //objdump can consume exe also

# For exploiting jmp-2-libc.

See [IPSEC’s October video](https://www.youtube.com/watch?v=K05mJazHhF4) , for reference

1. Use *info proc map* & *find* command in gdb, to know which third-party library you will be exploiting.
2. Get version of that library.

For libc , use *ldd --version* , for other use

*Dpkg -l <library\_name>*.

1. Now check if this library is using ASLR. For this run ldd <binary name> multiple times. Use below script,

*For i in `seq 0 to 20` ; do ldd <binary\_name> | grep -I <third-party library-name>; done*

This script will run 3rd party library 20 times. From address you can know if ASLR on and if there is any relation in the address.

1. Get offset for each address
   1. To get offset for system(), use

*readelf -s <full path of libc-library> | grep system*

* 1. To get offset for /bin/sh , use

*strings -a -t x < full path of libc-library > | grep “/bin/sh”*

1. To get address
   1. To get address of system(), use

*disassemble system()*

* 1. To get address of “/bin/sh”, use

*info proc map* & *find*  command.

1. Now you have all details required. All you need is to that the-addresses which you give coincide with address in libc. For this use, Step-3 to get relation between address. As your process ain’t crashing after each execution, you can brute-force ASLR. For this use your python python program