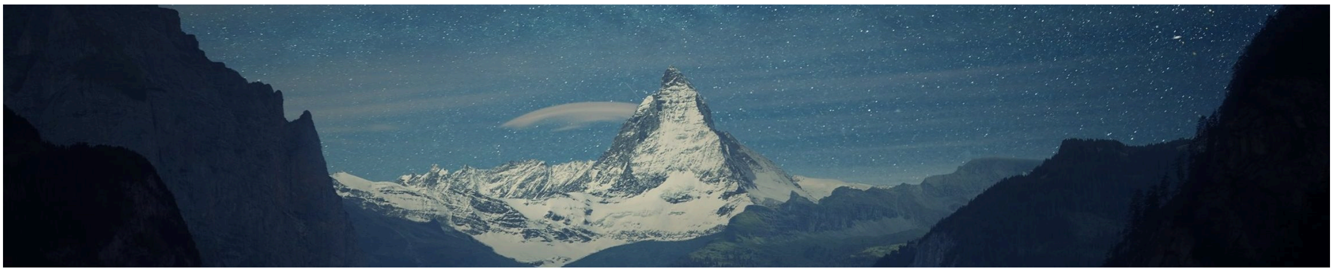
SillyRabbit's blog

路虽远行则将至，事虽难做则必成！



☰ 菜单

Patchelf和glibc-all-in_one使用笔记

2021年7月22日 作者 SillyRabbit

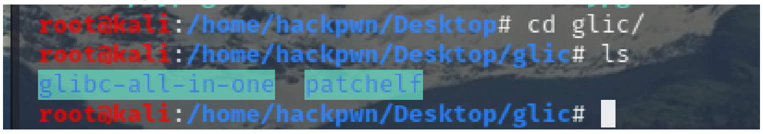
这个用来让程序链接指定的glibc，调试必备的。

先下载两个工具：

```
#Patchelf
git clone https://github.com/NixOS/patchelf

#glibc-all-in_one
git clone https://github.com/matrix1001/glibc-all-in-one
```

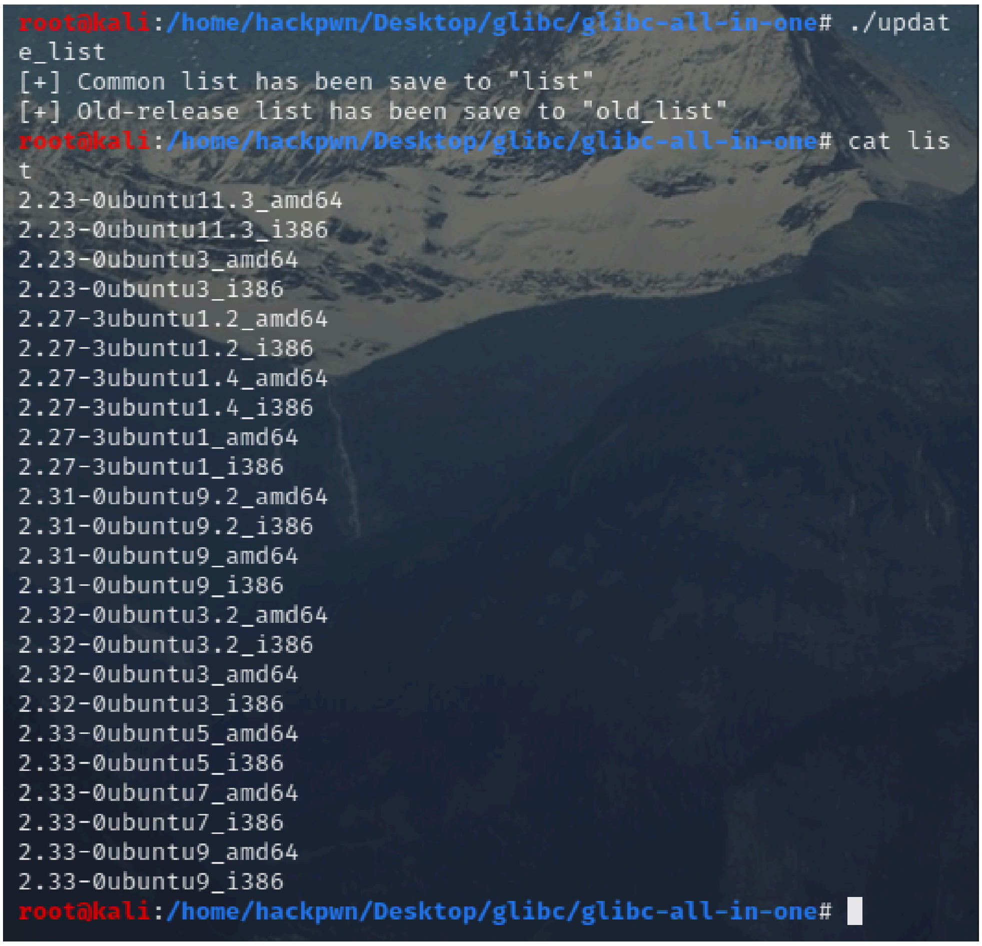
我创建了glibc目录，把两个工具放在里面



```
root@kali:/home/hackpwn/Desktop# cd glic/
root@kali:/home/hackpwn/Desktop/glic# ls
glibc-all-in-one  patchelf
root@kali:/home/hackpwn/Desktop/glic# 
```

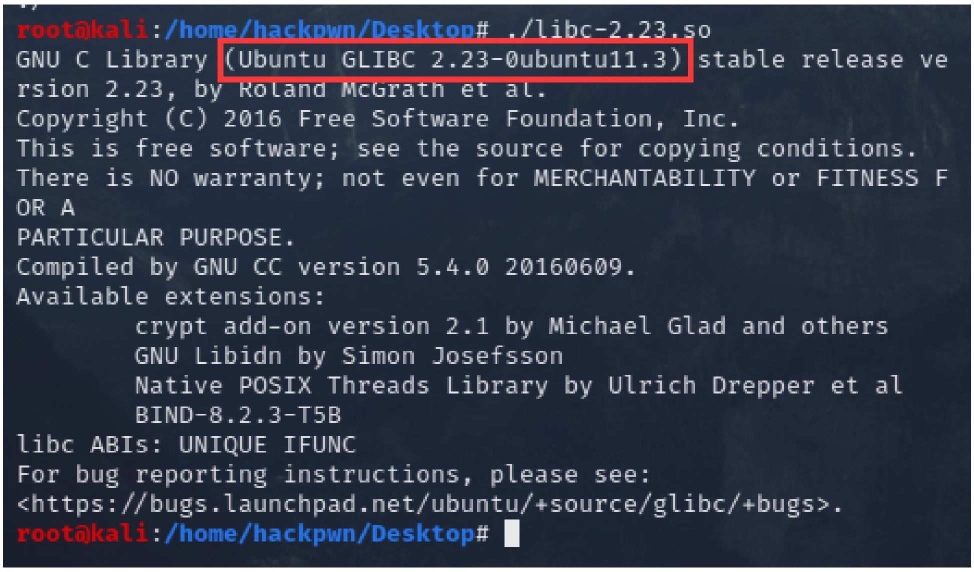
首先：

```
./update_list             #更新最新版本的glibc
cat list                 #查看可下载的glibc
```



```
root@kali:/home/hackpwn/Desktop/glibc/glibc-all-in-one# ./updat
e_list
[+] Common list has been save to "list"
[+] Old-release list has been save to "old_list"
root@kali:/home/hackpwn/Desktop/glibc/glibc-all-in-one# cat lis
t
2.23-0ubuntu11.3_amd64
2.23-0ubuntu11.3_i386
2.23-0ubuntu3_amd64
2.23-0ubuntu3_i386
2.27-3ubuntu1.2_amd64
2.27-3ubuntu1.2_i386
2.27-3ubuntu1.4_amd64
2.27-3ubuntu1.4_i386
2.27-3ubuntu1_amd64
2.27-3ubuntu1_i386
2.31-0ubuntu9.2_amd64
2.31-0ubuntu9.2_i386
2.31-0ubuntu9_amd64
2.31-0ubuntu9_i386
2.32-0ubuntu3.2_amd64
2.32-0ubuntu3.2_i386
2.32-0ubuntu3_amd64
2.32-0ubuntu3_i386
2.33-0ubuntu5_amd64
2.33-0ubuntu5_i386
2.33-0ubuntu7_amd64
2.33-0ubuntu7_i386
2.33-0ubuntu9_amd64
2.33-0ubuntu9_i386
root@kali:/home/hackpwn/Desktop/glibc/glibc-all-in-one# 
```

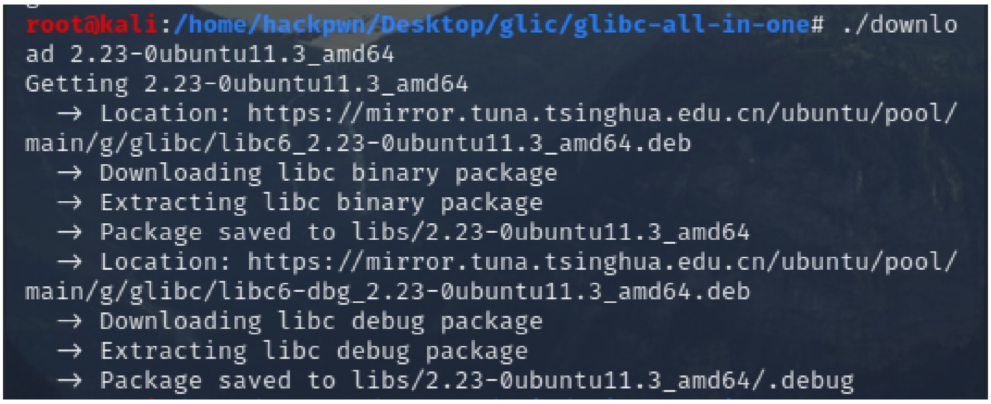
再根据题目所给的 libc， 找对应版本的连接器



```
root@kali:/home/hackpwn/Desktop# ./libc-2.23.so
GNU C Library (Ubuntu GLIBC 2.23-0ubuntu11.3) stable release ve
rsion 2.23, by Roland McGrath et al.
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS F
OR A
PARTICULAR PURPOSE.
Compiled by GNU CC version 5.4.0 20160609.
Available extensions:
        crypt add-on version 2.1 by Michael Glad and others
        GNU Libidn by Simon Josefsson
        Native POSIX Threads Library by Ulrich Drepper et al
        BIND-8.2.3-T5B
libc ABIs: UNIQUE IFUNC
For bug reporting instructions, please see:
<https://bugs.launchpad.net/ubuntu/+source/glibc/+bugs>.
root@kali:/home/hackpwn/Desktop# 
```

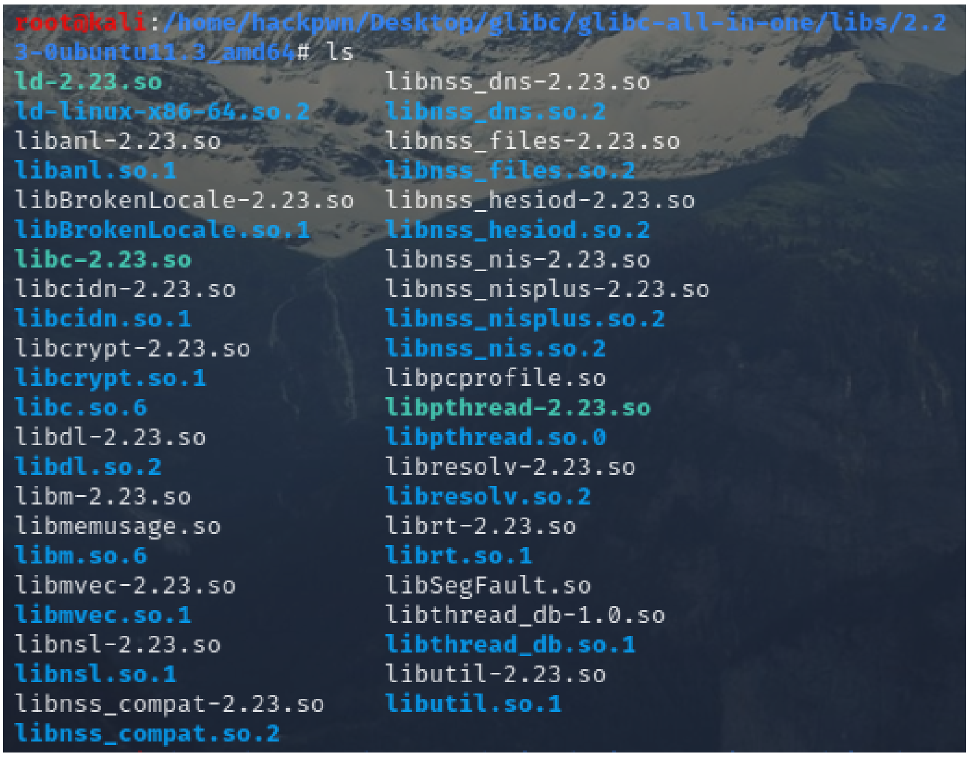
下载该连接器：

```
./download 2.23-0ubuntu11.3
```



```
root@kali:/home/hackpwn/Desktop/glic/glibc-all-in-one# ./downlo
ad 2.23-0ubuntu11.3_amd64
Getting 2.23-0ubuntu11.3_amd64
  -> Location: https://mirror.tuna.tsinghua.edu.cn/ubuntu/pool/
main/g/glibc/libc6_2.23-0ubuntu11.3_amd64.deb
  -> Downloading libc binary package
  -> Extracting libc binary package
  -> Package saved to libs/2.23-0ubuntu11.3_amd64
  -> Location: https://mirror.tuna.tsinghua.edu.cn/ubuntu/pool/
main/g/glibc/libc6-dbg_2.23-0ubuntu11.3_amd64.deb
  -> Downloading libc debug package
  -> Extracting libc debug package
  -> Package saved to libs/2.23-0ubuntu11.3_amd64/.debug
```

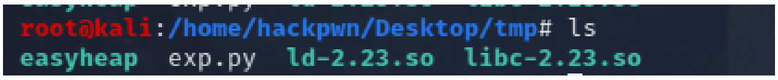
下载好的libc就在libs目录下



```
root@kali:/home/hackpwn/Desktop/glibc/glibc-all-in-one/libs/2.2
3-0ubuntu11.3_amd64# ls
ld-2.23.so               libnss_dns-2.23.so
ld-linux-x86-64.so.2     libnss_dns.so.2
libanl-2.23.so           libnss_files-2.23.so
libanl.so.1              libnss_files.so.2
libBrokenLocale-2.23.so  libnss_hesiod-2.23.so
libBrokenLocale.so.1     libnss_hesiod.so.2
libc-2.23.so             libnss_nis-2.23.so
libcidn-2.23.so          libnss_nisplus-2.23.so
libcidn.so.1             libnss_nisplus.so.2
libcrypt-2.23.so         libnss_nis.so.2
libcrypt.so.1            libpcprofile.so
libc.so.6                libpthread-2.23.so
libdl-2.23.so            libpthread.so.0
libdl.so.2               libresolv-2.23.so
libm-2.23.so             libresolv.so.2
libmemusage.so           librt-2.23.so
libm.so.6                librt.so.1
libmvec-2.23.so          libSegFault.so
libmvec.so.1             libthread_db-1.0.so
libnsl-2.23.so           libthread_db.so.1
libnsl.so.1              libutil-2.23.so
libnss_compat-2.23.so    libutil.so.1
libnss_compat.so.2
```

然后把ld文件和libc复制到pwn题目录下

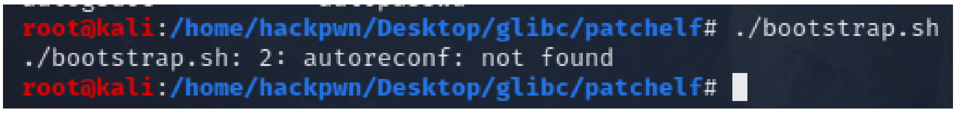
```
cp ./ld-2.23.so ../../../../tmp/
cp ./libc-2.23.so ../../../../tmp/
```



```
root@kali:/home/hackpwn/Desktop/tmp# ls
easyheap  exp.py  ld-2.23.so  libc-2.23.so
```

接下来使用Patchelf

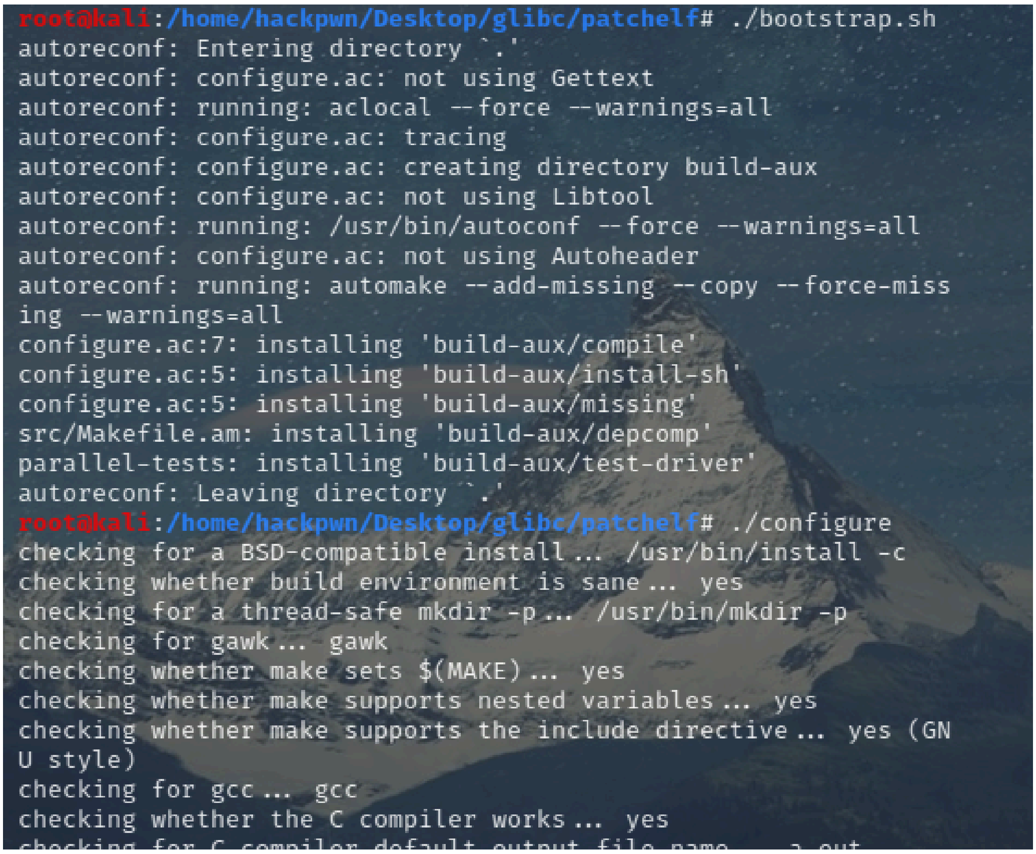
```
./bootstrap.sh
./configure
make
sudo make install
make check
```



```
root@kali:/home/hackpwn/Desktop/glibc/patchelf# ./bootstrap.sh
./bootstrap.sh: 2: autoreconf: not found
root@kali:/home/hackpwn/Desktop/glibc/patchelf# 
```

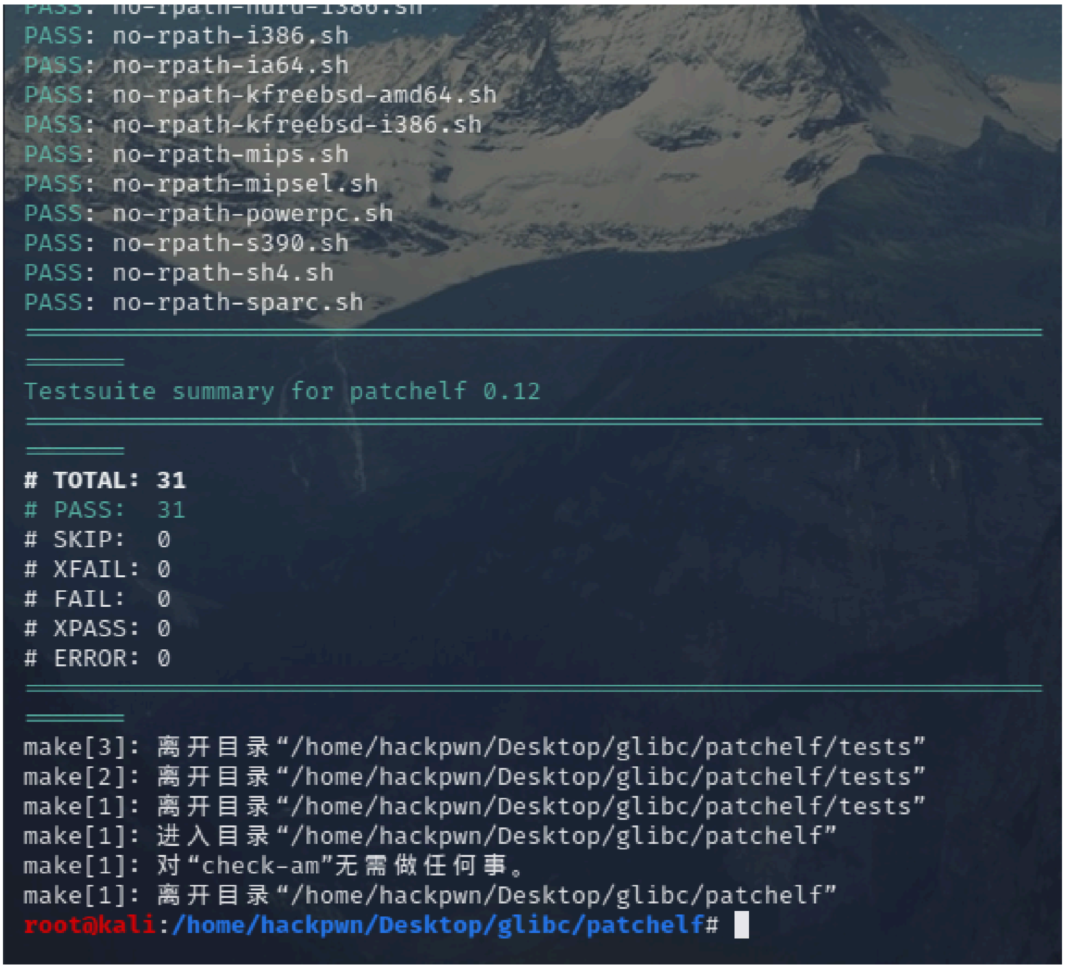
缺了autoreconf

```
apt-get install autoconf automake libtool
```



```
root@kali:/home/hackpwn/Desktop/glibc/patchelf# ./bootstrap.sh
autoreconf: Entering directory `.'
autoreconf: configure.ac: not using Gettext
autoreconf: running: aclocal --force --warnings=all
autoreconf: configure.ac: tracing
autoreconf: configure.ac: creating directory build-aux
autoreconf: configure.ac: not using Libtool
autoreconf: running: /usr/bin/autoconf --force --warnings=all
autoreconf: configure.ac: not using Autoheader
autoreconf: running: automake --add-missing --copy --force-miss
ing --warnings=all
configure.ac:7: installing 'build-aux/compile'
configure.ac:5: installing 'build-aux/install-sh'
configure.ac:5: installing 'build-aux/missing'
src/Makefile.am: installing 'build-aux/depcomp'
parallel-tests: installing 'build-aux/test-driver'
autoreconf: Leaving directory `.'
root@kali:/home/hackpwn/Desktop/glibc/patchelf# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether make supports the include directive... yes (GN
U style)
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
```

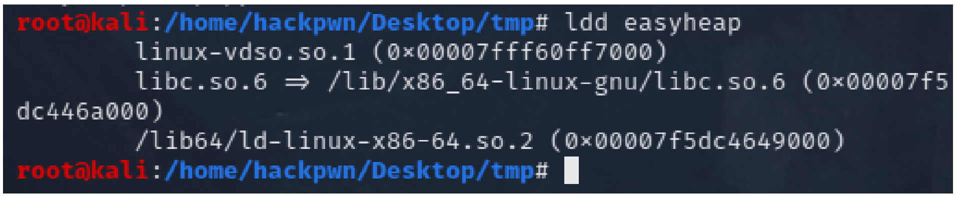
完成



```
PASS: no-rpath-hurd-i386.sh
PASS: no-rpath-i386.sh
PASS: no-rpath-ia64.sh
PASS: no-rpath-kfreebsd-amd64.sh
PASS: no-rpath-kfreebsd-i386.sh
PASS: no-rpath-mips.sh
PASS: no-rpath-mipsel.sh
PASS: no-rpath-powerpc.sh
PASS: no-rpath-s390.sh
PASS: no-rpath-sh4.sh
PASS: no-rpath-sparc.sh
============================================================================
Testsuite summary for patchelf 0.12
============================================================================
# TOTAL: 31
# PASS:  31
# SKIP:  0
# XFAIL: 0
# FAIL:  0
# XPASS: 0
# ERROR: 0
============================================================================
make[3]: 离开目录“/home/hackpwn/Desktop/glibc/patchelf/tests”
make[2]: 离开目录“/home/hackpwn/Desktop/glibc/patchelf/tests”
make[1]: 离开目录“/home/hackpwn/Desktop/glibc/patchelf/tests”
make[1]: 进入目录“/home/hackpwn/Desktop/glibc/patchelf”
make[1]: 对“check-am”无需做任何事。
make[1]: 离开目录“/home/hackpwn/Desktop/glibc/patchelf”
root@kali:/home/hackpwn/Desktop/glibc/patchelf# 
```

查看题目原来的 libc 和 ld

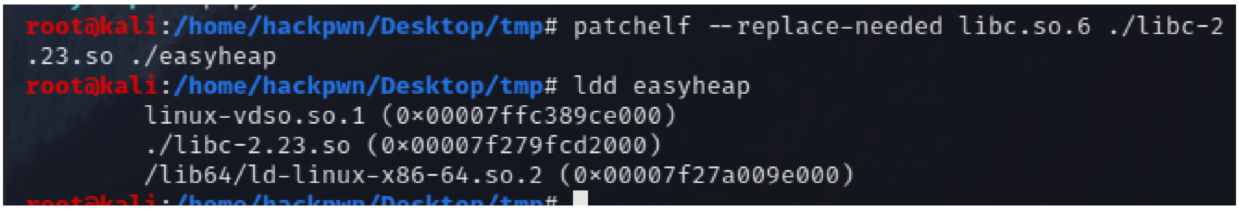
```
ldd easyheap
```



```
root@kali:/home/hackpwn/Desktop/tmp# ldd easyheap
        linux-vdso.so.1 (0x00007fff60ff7000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f5
dc446a000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f5dc4649000)
root@kali:/home/hackpwn/Desktop/tmp# 
```

替换libc

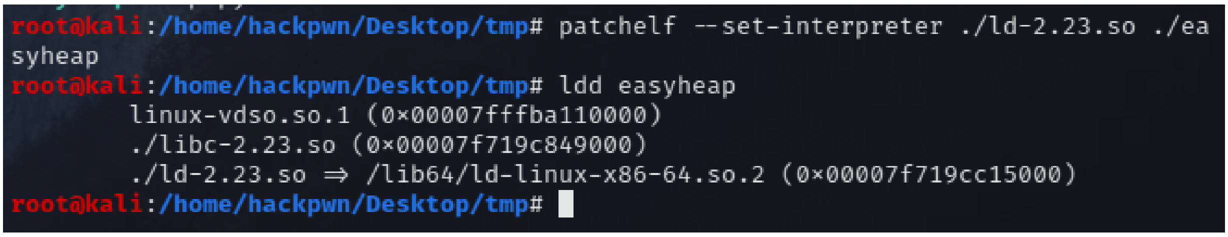
```
patchelf --replace-needed libc.so.6 ./libc-2.23.so ./easyheap
```



```
root@kali:/home/hackpwn/Desktop/tmp# patchelf --replace-needed libc.so.6 ./libc-2
.23.so ./easyheap
root@kali:/home/hackpwn/Desktop/tmp# ldd easyheap
        linux-vdso.so.1 (0x00007ffc389ce000)
        ./libc-2.23.so (0x00007f279fcd2000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f27a009e000)
root@kali:/home/hackpwn/Desktop/tmp# 
```

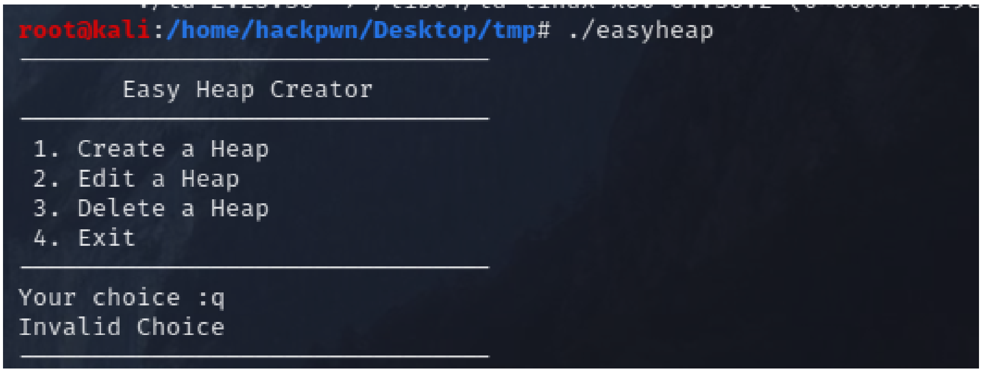
设置ld文件

```
patchelf --set-interpreter ./ld-2.23.so ./easyheap
```



```
root@kali:/home/hackpwn/Desktop/tmp# patchelf --set-interpreter ./ld-2.23.so ./ea
syheap
root@kali:/home/hackpwn/Desktop/tmp# ldd easyheap
        linux-vdso.so.1 (0x00007fffba110000)
        ./libc-2.23.so (0x00007f719c849000)
        ./ld-2.23.so => /lib64/ld-linux-x86-64.so.2 (0x00007f719cc15000)
root@kali:/home/hackpwn/Desktop/tmp# 
```

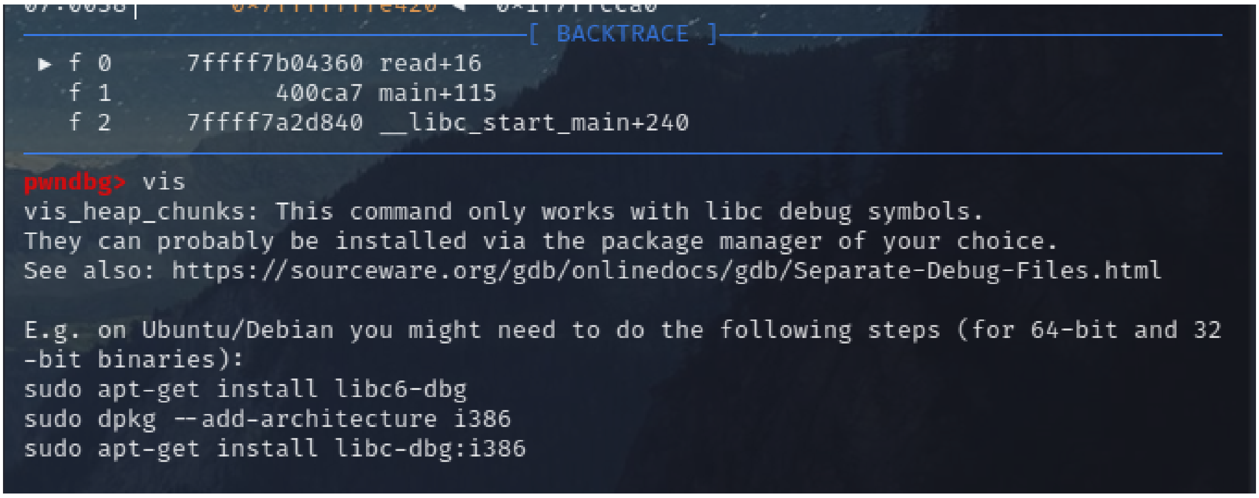
正常执行



```
root@kali:/home/hackpwn/Desktop/tmp# ./easyheap
--------------------------------
       Easy Heap Creator
--------------------------------
 1. Create a Heap
 2. Edit a Heap
 3. Delete a Heap
 4. Exit
--------------------------------
Your choice :q
Invalid Choice
--------------------------------
```

gdb调试

直接整会出问题



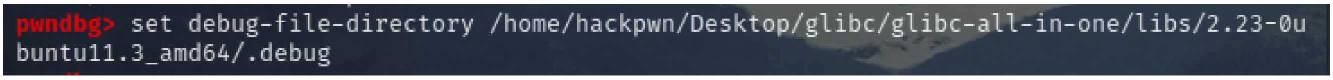
```
──────────────────────────────────────[ BACKTRACE ]──────────────────────────────────────
 ► f 0     7ffff7b04360 read+16
   f 1           400ca7 main+115
   f 2     7ffff7a2d840 __libc_start_main+240
─────────────────────────────────────────────────────────────────────────────────────────
pwndbg> vis
vis_heap_chunks: This command only works with libc debug symbols.
They can probably be installed via the package manager of your choice.
See also: https://sourceware.org/gdb/onlinedocs/gdb/Separate-Debug-Files.html

E.g. on Ubuntu/Debian you might need to do the following steps (for 64-bit and 32
-bit binaries):
sudo apt-get install libc6-dbg
sudo dpkg --add-architecture i386
sudo apt-get install libc-dbg:i386
```

有设置一下符号表和将.debug，移动到题目的同级目录下两个办法

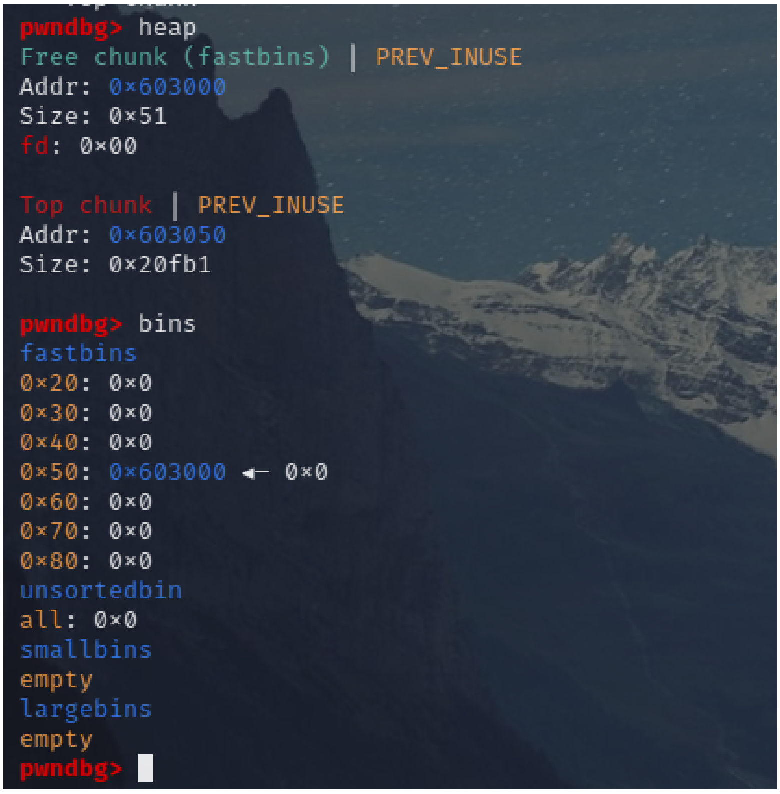
```
set debug-file-directory /home/hackpwn/Desktop/glibc/glibc-all-in-one/libs/2.23-0ubu
```

在gdb里面设置调试符号表，路径在libs里面



```
pwndbg> set debug-file-directory /home/hackpwn/Desktop/glibc/glibc-all-in-one/libs/2.23-0u
buntu11.3_amd64/.debug
```

heap，bins，vis等正常执行



```
pwndbg> heap
Free chunk (fastbins) | PREV_INUSE
Addr: 0x603000
Size: 0x51
fd: 0x00

Top chunk | PREV_INUSE
Addr: 0x603050
Size: 0x20fb1

pwndbg> bins
fastbins
0x20: 0x0
0x30: 0x0
0x40: 0x0
0x50: 0x603000 ◂— 0x0
0x60: 0x0
0x70: 0x0
0x80: 0x0
unsortedbin
all: 0x0
smallbins
empty
largebins
empty
pwndbg> 
```

pwntools中gdb调试

只能是将.debug，移动到题目的同级目录下

```
mkdir .debug
cp /home/hackpwn/Desktop/.../.debug/* ./
```

亲测有效

Post Views: 2,351

Pwn、 小结

笔记、 调试

‹ 堆入门5.1—Fastbin Attack

› Fastbin Attck—Double Free

发表评论

名称 *

电子邮箱地址 *

网站地址

□ 在此浏览器中保存我的显示名称、邮箱地址和网站地址，以便下次评论时使用。

发表评论

此站点使用Akismet来减少垃圾评论。了解我们如何处理您的评论数据。

分类

选择分类

标签

Canary csu CVE复现 Fastbin Attack Got劫持 heap hook house of einherjar House Of Orange house_of_roman how2heap large bin attack Metasploit Nmap off_by_null off_by_one one_gadget Python realloc shell shellcode Sqlmap stack Tcache UAF Unlink Unsortedbin Attack wp XXE 《程序员的自我修养》 不知道打什么标签系列 反序列化 后续 堆入门 存储介质 学习思考 应付作业系列 整型溢出 月饼杯 栈迁移 格式化字符串 环境搭建 笔记 系统调用 调试

近期文章

- 信息搜集小Tip
- XXE简要学习笔记
- 取证理论笔记——文件系统结构（FAT）
- 取证理论笔记——介质存储机制
- 免杀学习笔记（概念）