Operating System

Version	
Kernel	Linux 5.13.0-51-generic (x86_64)
Version	#58~20.04.1-Ubuntu SMP Tue Jun 14 11:29:12 UTC 2022
C Library	GNU C Library / (Ubuntu GLIBC 2.31-0ubuntu9.9) 2.31
Distribution	Ubuntu 20.04.4 LTS
Current Session	
Computer Name	manohar-Victus
User Name	manohar (NAGA MANOHAR)
Language	en_IN (en_IN:en)
Home Directory	/home/manohar
Misc	
Uptime	1 minute
Load Average	1.24, 0.53, 0.20
Available entropy in /dev/random	1795 bits (medium)

Kernel Modules

Loaded Modules	
rfcomm	Bluetooth RFCOMM ver 1.11
xt_conntrack	Xtables: connection tracking state match
xt_MASQUERADE	Xtables: automatic-address SNAT
nfnetlink	Netfilter messages via netlink socket

Filesystems

Mounted File Systems		
udev	/dev	0.00 % (3.5 GiB of 3.5 GiB)
tmpfs	/run	0.29 % (726.2 MiB of 728.3 MiB)
/dev/nvme0n1p5	/	36.63 % (38.8 GiB of 61.2 GiB)
tmpfs	/dev/shm	0.00 % (3.6 GiB of 3.6 GiB)
tmpfs	/run/lock	0.08 % (5.0 MiB of 5.0 MiB)
tmpfs	/sys/fs/cgroup	0.00 % (3.6 GiB of 3.6 GiB)
/dev/loop0	/snap/core/13250	100.00 % (0.0 B of 111.8 MiB)
tmpfs	/run/snapd/ns	0.29 % (726.2 MiB of 728.3 MiB)

Processor

Processors			
Package Information			
AMD Ryzen 5 5600H with Radeon Graphics	1	0:0	3300.00 MHz

Memory

Memory		
MemTotal	Total Memory	7458028 KiB
MemFree	Free Memory	4640992 KiB
MemAvailable		5892896 KiB
Buffers		104644 KiB
Cached		1307936 KiB
SwapCached	Cached Swap	0 KiB
SwapTotal	Virtual Memory	2097148 KiB
SwapFree	Free Virtual Memory	2097148 KiB

PCI Devices

PCI Devices	
Host bridge	Advanced Micro Devices, Inc. [AMD] Renoir Root Complex
IOMMU	Advanced Micro Devices, Inc. [AMD] Renoir IOMMU
Host bridge	Advanced Micro Devices, Inc. [AMD] Renoir PCIe Dummy Host Bridge
PCI bridge	Advanced Micro Devices, Inc. [AMD] Renoir PCIe GPP Bridge (prog-if 00 [Normal decode])
USB controller	Advanced Micro Devices, Inc. [AMD] Renoir USB 3.1 (prog-if 30 [XHCI])
Multimedia controller	Advanced Micro Devices, Inc. [AMD] Raven/Raven2/FireFlight/Renoir Audio Processor (rev 01)
Audio device	Advanced Micro Devices, Inc. [AMD] Family 17h (Models 10h-1fh) HD Audio Controller

USB Devices

USB Devices
Linux Foundation 3.0 root hub
Lite-On Technology Corp. Dell Wireless Device
Linux Foundation 2.0 root hub
Linux Foundation 3.0 root hub
Realtek Semiconductor Corp. Bluetooth Radio
Chicony Electronics Co., Ltd HP Wide Vision HD Camera
Linux Foundation 2.0 root hub

Battery

Battery: BAT0	
State	Discharging
Capacity	54 / Normal
Battery Technology	Li-ion
Manufacturer	HP
Model Number	Primary

Sensors

Sensors		
../../thermal_zone0/temp1	Temperature	50.00°C
../../BAT0/in0	Voltage	14.75V
../../nvme0/temp1	Temperature	31.85°C
../../nvme0/temp2	Temperature	31.85°C
../../nvme0/temp3	Temperature	33.85°C
amdgpu/temp1	Temperature	40.00°C
amdgpu/in0	Voltage	1.37V
amdgpu/in1	Voltage	0.69V
thermal/thermal_zone0	Temperature	50.00°C

STORAGE:

SSD	512 GB
HDD	NIL
RAM	8GB(2 slots)

DMI

Product	
Name	Victus by HP Laptop 16-e0xxx (Hewlett-Packard, www.hp.com)
Family	103C_5335M7 HP VICTUS (Hewlett-Packard, www.hp.com)
Vendor	HP (Hewlett-Packard, www.hp.com)
Version	(Not available; Perhaps try running HardInfo as root.)
BIOS	
Date	07/12/2021
Vendor	AMI
Version	F.05
Board	
Name	88EB
Vendor	HP (Hewlett-Packard, www.hp.com)
Version	80.23
Serial Number	(Not available; Perhaps try running HardInfo as root.)
Asset Tag	Base Board Asset Tag
Chassis	
Vendor	HP (Hewlett-Packard, www.hp.com)
Type	[10] Notebook
Version	Chassis Version
Serial Number	(Not available; Perhaps try running HardInfo as root.)
Asset Tag	(Not available; Perhaps try running HardInfo as root.)

Benchmarks

GPU Drawing

GPU Drawing		
AMD Ryzen 5 5600H with Radeon Graphics	12x 3300.00 MHz	7859.28

CPU Zlib

CPU Zlib		
AMD Ryzen 5 5600H with Radeon Graphics	12x 3300.00 MHz	2.18
PowerPC 740/750	1x 280.00 MHz	2150.60

CPU CryptoHash

CPU CryptoHash		
AMD Ryzen 5 5600H with Radeon Graphics	12x 3300.00 MHz	1078.35

CS0-Q5-A2

Q5 (a)

The interpreter used to generate 'q5.out' was [./lib6-amd64_2.27-3ubuntu1_i386.ld] but the interpreter in my system is different [/lib64/ld-linux-x86-64.so.2] hence the file can't be decoded to generate correct output

If the interpreter is changed to [/lib64/ld_linux-x86-64.so.2] by "patchelf --set-interpreter q5.out /lib64/ld-linux_x86-64.so.2" command

we get correct output

(Q5(b)

The binary file is expected to be executed by Machine: "Advanced Micro devices X86-64"

It provides Information about:

(1) Program Headers
(2) ELF Headers
(3) Section Headers
(4) Processor Architecture
(5) Interpreter
(6) OS

2021101128

assemblycode (0xc, 0x15)
↳ 2 parameters

↓

This will do ↳ push 0x15
push 0xc

the assemblycode is called => as call assemblycode

---

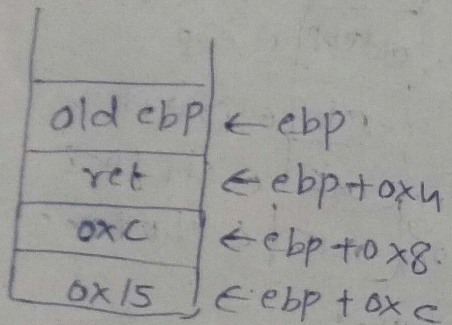
assemblycode:

<+0> push ebp # push ebp onto stack

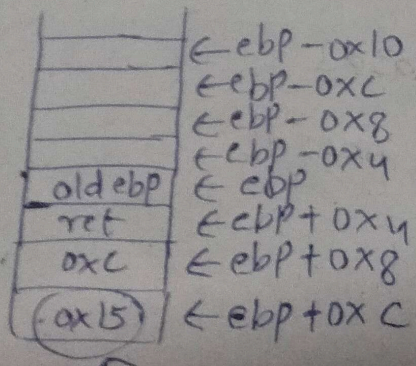
<+1> mov ebp, esp # ebp is given the value of esp (address that esp contains)

Stack status after <+0>, <+1>

=>



<+3>: sub esp, 0x10 [0x10 = 16 = 4 bytes of ~~memory~~ space is made for variables]



=> new space created for 4 variables
↳ stack grows ~~up here!~~ to lower addresses.

<+6>: mov eax, DWORD PTR [ebp+0xc]

Here the instructions are reverse => This value is moved to eax register

=> eax = 0x15 = ebp+0xc

<+9>: mov DWORDPTR[ebp-0x4], eax

variable_1 = [ebp-0x4] = eax = 0x15 (previously changed in <+4>)
(created by sub)

<+12>: mov eax, DWORDPTR [ebp+0x8]

mov value (ebp+0x8) = 0xc => eax = 0xc

<+15>: mov DWORDPTR [ebp-0x8], eax

store eax in a new variable at [ebp-0x8] = variable_2 = 0xc

<+18>: jmp 0x50c <asm2+31>
↳ jump to <+31>

These are Local to this stack frame {
variable_4 ← ebp-0x10
variable_3 ← ebp-0xc
variable_2 ← ebp-0x8
variable_1 ← ebp-0x4
}

Stack Status:

	← ebp-0x10
	← ebp-0xc
0xc	← ebp-0x8
0x15	← ebp-0x4
oldebp	← ebp
ret	← ebp+0x4
0xc	← ebp+0x8
0x15	← ebp+0xc

<+18> jumps to <+31>

hence <+20> add DWORDPTR [ebp-0x4], 0x
<+24> add DWORDPTR [ebp-0x8], 0xa

are not executed

<+31>: cmp DWORDPTR [ebp-0x8], 0xa3d3

Compares variable_2 [ebp-0x8], with 0xa3d3
has (0xc) in it

<+38>: jle 0x501 <asm2+20>

condition true => [ebp-0x8] = (0xc <= 0xa3d3)

=> jump to line <+20>

<+20> add

=> v

<+24> add

=>

Now <+

Thus <+

<+40>:

eax = va

ac 0x

=> <+

are

ag

Hence a

<+20> add DWORD PTR [ebp-0x4], 0x1

variable_1 = variable_1 + 1

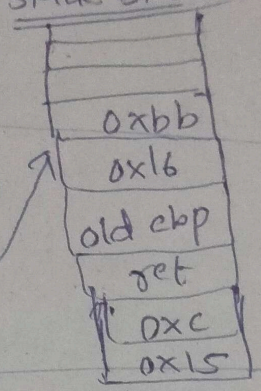
<+24> add DWORD PTR [ebp-0x8], 0xaf

variable_2 = variable_2 + 0xaf

= 0xc + 0xaf

= 0xbb

stack status :



as 0xbb < 0xa3d3 => condition true in <+38>

again jump back to <+20>

=> <+20>, <+24>, <+31>, <+38> are executed again and again until, variable_2 > 0xa3d3

Hence this happens after 240 iterations

and by then

variable_1 = variable_1 + 240 (decimal)

=> variable_1 = 0x105 (hexa-decimal)

<+40>: mov eax, DWORD PTR [ebp-0x4]

eax = variable_1 = 0x105 (hex-decimal)

<+43>: leave

This commands pops the stack frame and old frame pointer is given to ebp

=> stack space is cleared

=> NO seg-fault occurs

<+44>: ret # returns the latest eax value that is 0x105

Hence, assembly code function returns 0x105