Week3- Write-up

Vishnu Vardhan Ciripuram
N14912012
vc2499

Challenge: Knapsack

The challenge involved solving a linear equation involving item quantities to satisfy a specific constraint, allowing us to retrieve the flag.

Solution Steps:

- I opened the binary file in **Binary Ninja** and analyzed the main function which handled the input validation and called another function called process.

```
000012b6  int32_t main(int32_t argc, char** argv, char** envp)

000012c2      int32_t argc_1 = argc
000012c5      char** argv_1 = argv
000012c9      void* fsbase
000012c9      int64_t rax = *(fsbase + 0x28)
000012dd      initialize()
000012ec      puts(str: "\n\n\tLet's go shopping!\n")
00001300      printf(format: "\tHow many of each would you lik…  ")
0000133a      int32_t var_28
0000133a      int32_t var_24
0000133a      int32_t var_20
0000133a      int32_t var_1c
0000133a      int32_t var_18
0000133a      int32_t var_14
0000133a      __isoc99_scanf(format: "%d - %d - %d - %d - %d - %d", &var_28, &var_24, &var_20, &var_1c, &var_18, &var_14)
0000136b      int32_t result
0000136b
0000136b      if (var_28 s< 0 || var_24 s< 0 || var_20 s< 0 || var_1c s< 0 || var_18 s< 0 || var_14 s< 0)
00001377          puts(str: "\tNah, that's not how we count a…  ")
0000137c          result = 1
0000136b      else if (process(var_28, var_24, var_20, var_1c, var_18, var_14) == 0)
000013d1          puts(str: "\n\tNope! Try again friend!\n")
000013d6          result = 1
000013a5      else
000013b1          puts(str: "\n\tYou made great choices!")
000013bb          get_flag()
000013c0          result = 0
000013c0
000013df      *(fsbase + 0x28)
000013df
000013e8      if (rax == *(fsbase + 0x28))
000013f0          return result
000013f0
000013ea      __stack_chk_fail()
000013ea      noreturn
```

- The process function was responsible for validating the user's input by checking a **linear equation** involving six arguments against a target value using specific coefficients.
- Inside the process function, I identified the following linear constraint:
- arg6×var6+arg1×var1+arg2×var2+arg3×var3+arg4×var4+arg5×var5=var0

```
00001229  int64_t process(int32_t arg1, int32_t arg2, int32_t arg3, int32_t arg4, int32_t arg5, int32_t arg6)

00001294      if (arg6 s<= 0)
00001296          return 0
00001296
000012a6      if (var_6 * arg6 + var_1 * arg1 + var_2 * arg2 + var_3 * arg3 + var_4 * arg4 + var_5 * arg5 != var_0)
000012af          return 0
000012af
000012a8      return 1
```

- To solve this, I needed to know the values of the coefficients (var6, var1, var2, var3, var4, var5) and the target value (var0).
- I explored the .data section in Binary Ninja, where I found the following **hexadecimal** values:
    - var0 = 0x645, var1 = 0xd7, var2 = 0x113, var3 = 0x14f, var4 = 0x163, var5 = 0x1a4, var6 = 0x244

```
.data (PROGBITS) section started  {0x4000-0x402c}
00004000  __data_start:
00004000  00 00 00 00 00 00 00 00

00004008  void* __dso_handle = __dso_handle
00004010  uint32_t var_0 = 0x645
00004014  uint32_t var_1 = 0xd7
00004018  uint32_t var_2 = 0x113
0000401c  uint32_t var_3 = 0x14f
00004020  uint32_t var_4 = 0x163
00004024  uint32_t var_5 = 0x1a4
00004028  uint32_t var_6 = 0x244
.data (PROGBITS) section ended  {0x4000-0x402c}
```

- I then converted these hexadecimal values to their **decimal equivalents**:
    - var0 = 1605, var1 = 215, var2 = 275, var3 = 335, var4 = 355, var5 = 420, var6 = 580
- With these values, the equation became:
    - 580×arg6+215×arg1+275×arg2+335×arg3+355×arg4+420×arg5=1605
- I used the **Z3 Solver** in Python to find values for arg1 through arg6 that satisfied the equation.
- After setting up the constraints and solving, I found the solution:
    - arg1 = 0, arg2 = 0, arg3 = 2, arg4 = 1, arg5 = 0, arg6 = 1
- I formatted the solution in the expected format:
    - 0 - 0 - 2 - 1 - 0 - 1
- I submitted this to the server and received the flag: **flag{1ts_n0t_t0_b4d_s0lv1ng_pr0bl3ms_w1th_Z3!_f49a0a291edbcae9}**

```
┌──(kali㉿kali)-[~/Downloads/offsec]
└─$ python3 knapsack.py
[+] Opening connection to offsec-chalbroker.osiris.cyber.nyu.edu on port 1260: Done
Please input your NetID (something like abc123):
hello, vc2499. Please wait a moment ...


        Let's go shopping!

        How many of each would you like?
Solution: 0 - 0 - 2 - 1 - 0 - 1
[+] Receiving all data: Done (122B)
[*] Closed connection to offsec-chalbroker.osiris.cyber.nyu.edu port 1260
Server Response:


        You made great choices!

        Here's your flag, friend: flag{1ts_n0t_t0_b4d_s0lv1ng_pr0bl3ms_w1th_Z3!_f49a0a291edbcae9}
```

Script:

knapsack.py
~/Downloads/offsec

```python
from pwn import *
from z3 import *

r = remote('offsec-chalbroker.osiris.cyber.nyu.edu', 1260)
print(r.recvuntil(b'NetID (something like abc123): ').decode())
r.sendline(b'vc2499')
print(r.recvuntil(b'How many of each would you like? ').decode())

args = [Int(f'arg{i}') for i in range(1, 7)]
solver = Solver()
solver.add(Sum([c * arg for c, arg in zip([215, 275, 335, 355, 420, 580], args)]) == 1605, *[arg >= 0 for arg in args])

model = solver.model() if solver.check() == sat else exit("No solution found.")
solution = " - ".join(str(model[arg]) for arg in args)
print(f"Solution: {solution}")

r.sendline(solution.encode())
print(f"Server Response:\n{r.recvall().decode()}")
```

Challenge: Disks Game

The challenge involved determining the correct number of disks for a recursive function such that the total number of moves matched a target value (goal).

Solution Steps:

- I opened the binary file in Binary Ninja and focused on the main function where I found that it calls the process function.

```
00001300  int32_t main(int32_t argc, char** argv, char** envp)

0000130c      int32_t argc_1 = argc
0000130f      char** argv_1 = argv
00001318      initialize()
0000132c      printf(format: "\nHow many disks do you want to …  ")
00001336      char rax_3 = get_number()
00001336
0000134b      if (process(zx.d(rax_3)) != 0)
0000135c          printf(format: "\nGood job! You selected the rig…  ")
00001366          get_flag()
0000136b          return 0
0000136b
00001387      printf(format: "\nNah, %u is not the number I ha…  ", zx.q(rax_3))
00001396      puts(str: "Try again, friend!\n\n")
0000139b      return 1
```

- The process function called a recurse function to compute total_moves based on the number of disks (arg1).
- The process function also validated whether total_moves matched the goal value .

```
000012aa  int64_t process(int32_t arg1)

000012bd      if (arg1 s<= 0)
000012bf          return 0
000012bf
000012da      recurse(arg1, 0x53, 0x54, 0x41)
000012da
000012f0      if (total_moves != goal)
000012f9          return 0
000012f9
000012f2      return 1
```

- I found the goal value in the **initialize** function which is **0x7fffffff (2147483647 when converted to binary)**

```
000014b2  int64_t initialize()

000014d3      setvbuf(fp: stdout, buf: nullptr, mode: 2, size: 0)
000014f1      int64_t result = setvbuf(fp: stdin, buf: nullptr, mode: 2, size: 0)
000014f6      goal = 0x7fffffff
00001503      return result
```

- The **recurse function** made two recursive calls and incremented total_moves each time arg1 was not zero and the growth of total_moves followed an exponential pattern with each increase in arg1.

```
00001249  void recurse(int32_t arg1, int32_t arg2, int32_t arg3, int32_t arg4)

00001265      if (arg1 != 0)
00001278          recurse(zx.q(arg1 - 1), zx.q(arg2), zx.q(arg4), zx.q(arg3))
0000128e          recurse(zx.q(arg1 - 1), zx.q(arg4), zx.q(arg3), zx.q(arg2))
0000129e          total_moves += 1
```

- **I** created a **python script** to simulate the recursive behavior using the exact logic found in the recurse function.
- I ran the script and found that 31 disks resulted in 2147483647 moves.

```
┌──(kali㉿kali)-[~/Downloads/offsec]
└─$ python3 disks_game.py
No of disks: 1, Total Moves: 1
No of disks: 2, Total Moves: 3
No of disks: 3, Total Moves: 7
No of disks: 4, Total Moves: 15
No of disks: 5, Total Moves: 31
No of disks: 6, Total Moves: 63
No of disks: 7, Total Moves: 127
No of disks: 8, Total Moves: 255
No of disks: 9, Total Moves: 511
No of disks: 10, Total Moves: 1023
No of disks: 11, Total Moves: 2047
No of disks: 12, Total Moves: 4095
No of disks: 13, Total Moves: 8191
No of disks: 14, Total Moves: 16383
No of disks: 15, Total Moves: 32767
No of disks: 16, Total Moves: 65535
No of disks: 17, Total Moves: 131071
No of disks: 18, Total Moves: 262143
No of disks: 19, Total Moves: 524287
No of disks: 20, Total Moves: 1048575
No of disks: 21, Total Moves: 2097151
No of disks: 22, Total Moves: 4194303
No of disks: 23, Total Moves: 8388607
No of disks: 24, Total Moves: 16777215
No of disks: 25, Total Moves: 33554431
No of disks: 26, Total Moves: 67108863
No of disks: 27, Total Moves: 134217727
No of disks: 28, Total Moves: 268435455
No of disks: 29, Total Moves: 536870911
No of disks: 30, Total Moves: 1073741823
No of disks: 31, Total Moves: 2147483647
The number of disks is: 31
```

- I submitted the value 31 to the server and received the flag: **flag{1ts_n0t_t0_b4d_s0lv1ng_pr0bl3ms_w1th_r3curs10n!}**

```
┌──(kali㉿kali)-[~/Downloads/offsec]
└─$ nc offsec-chalbroker.osiris.cyber.nyu.edu 1261
Please input your NetID (something like abc123): vc2499
hello, vc2499. Please wait a moment ...

How many disks do you want to start with?
> 31

Good job! You selected the right number of disks!
Here's your flag, friend: flag{r3curs1v3_funct10ns_4nd_3xp0n3nt14l_gr0wth!_c3c7c999b5f3a4ed}
```

Script:

Open

disks_game.py
~/Downloads/offsec

```python
goal = 2147483647

def recurse(arg1, arg2, arg3, arg4):
    global total_moves
    if arg1:
        recurse(arg1 - 1, arg2, arg4, arg3)
        recurse(arg1 - 1, arg4, arg3, arg2)
        total_moves += 1

for arg1 in range(1, 50):
    total_moves = 0
    recurse(arg1, 0x53, 0x54, 0x41)
    print(f"No of disks: {arg1}, Total Moves: {total_moves}")
    if total_moves == goal:
        print(f"The number of disks is: {arg1}")
        break

```