HOMEWORK 6

~Purva Naresh Rumde (pr23b)

Problem ID	Captured Flag	Steps
P1	**FSUctf{l1ik3_c4tch1ng_1f3rfl135_1n_4_j4r}**	In this problem I unzipped the challenge.jar file and it consisted of 5 files. I opened every file in ghidra and got half strings that were together forming the flag. So after reversing a few and getting them all I got the flag.
P2	**fsuctf{p4ck3d_full_0f_str1ng5}**	In this problem, I figured out that UPX was used so I unpacked with it and got the new file. Then I processed that file in ghidra I went through the main function and discovered the two functions op1 and op2. So I reversed the code for op2 and gave the output as input and got the original input and then reversed it to get the flag.
P3	**FSUctf{why_u_50_4ngry_b01_0r_g1rl}**	In this problem I created an angr script for getting the flag. I first installed angr on my kali and had to install all the necessary packages.

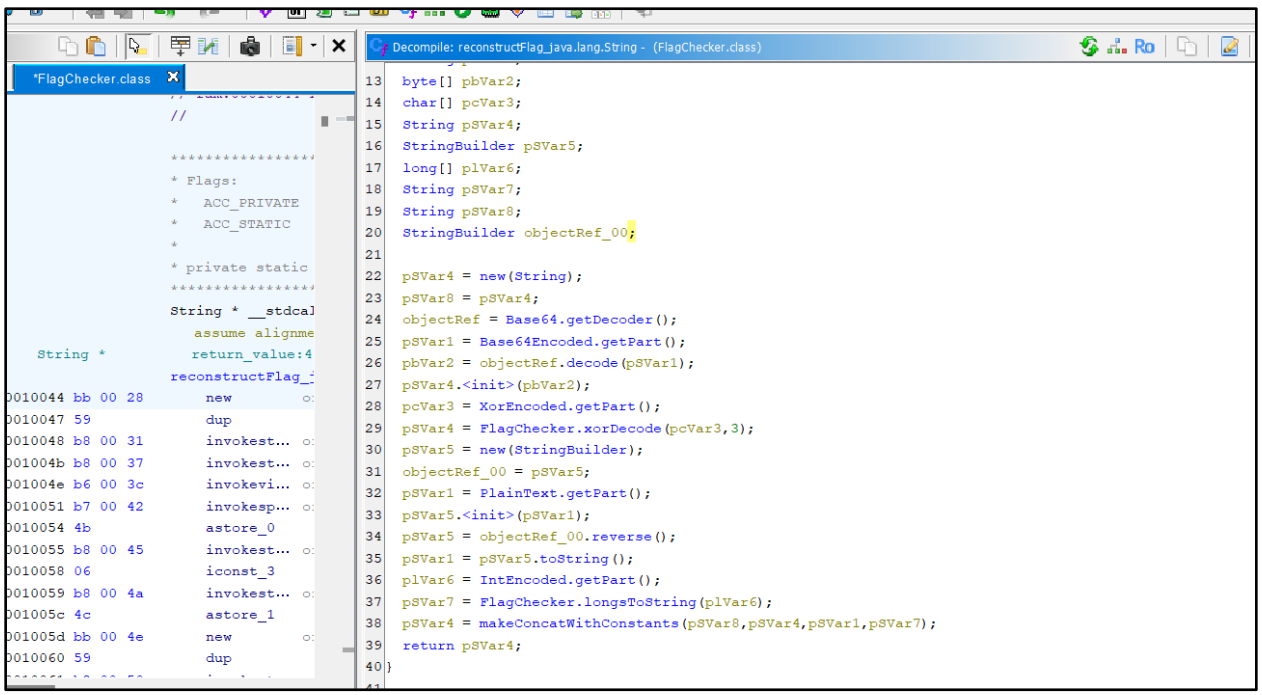
Q1.

In this problem we are provided with challenger.jar file that I unzipped using a java command and got the following 5 files:

So after looking at the meta data information I figured out that FlagChecker is the main file.

Here you can see that a string from XorEncoding is taken as first argument and 3 is taken as second argument and both are passed through xorDecode function which is inside the FlagChecker class.

After doing that I got the string: “3_c4tch1ng”

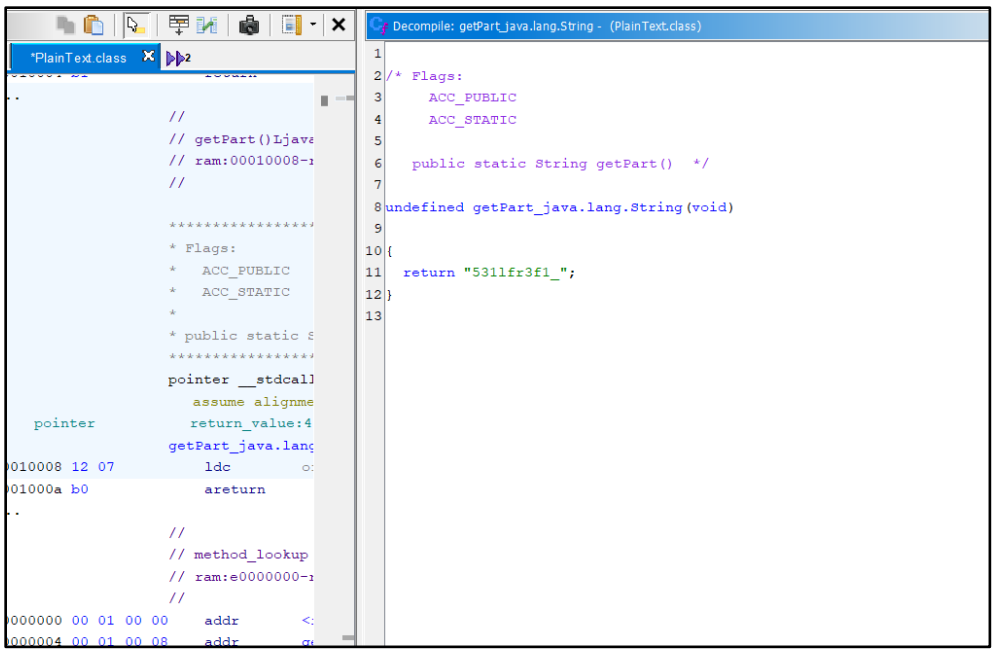


```
Decompile: getPart_char[] - (XorEncoded.class)

7
8 char[] getPart_char[](void)
9
10 {
11   int iVar1;
12   char cVar2;
13   String objectRef;
14   char[] pcVar3;
15   int iVar4;
16   int iVar5;
17   char[] pcVar6;
18
19   objectRef = "0\\`7w`k2md";
20   iVar4 = objectRef.length();
21   pcVar3 = new char[iVar4];
22   iVar4 = 0;
23   while( true ) {
24     iVar5 = iVar4;
25     iVar1 = objectRef.length();
26     if (iVar1 <= iVar5) break;
27     iVar5 = iVar4;
28     pcVar6 = pcVar3;
29     cVar2 = objectRef.charAt(iVar4);
30     pcVar6[iVar5] = cVar2;
31     iVar4 = iVar4 + 1;
32   }
33   return pcVar3;
34 }
35
```

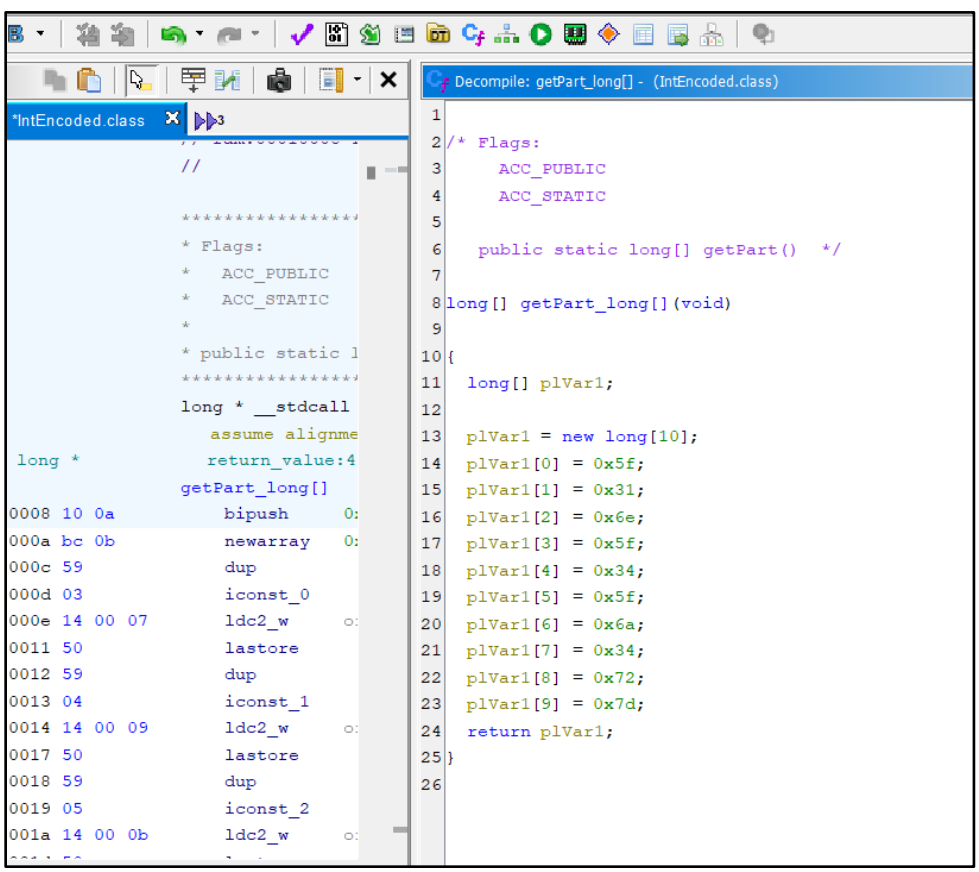
Now lets move to Base64Encoded file. This file consists of a string “RlNVY3Rme2wxaWs=” that I then decoded with from base64 and got the answer as : “FSUctf{l1ik”

```
Listing: Base64Encoded.class

                    assume alignmentPad = 0x3
undefined *         return_value:4<RETURN>
                    getPart_java.lang.String
ram:00010008 12 07        ldc        offset CPOOL[7]
ram:0001000a b0           areturn
...........
                    //
                    // method_lookup
                    // ram:e0000000-ram:e003ffff
                    //
ram:e0000000 00 01 00 00  addr       <init>_void
ram:e0000004 00 01 00 08  addr       getPart_java.lang
ram:e0000008 ff           ??         FFh
ram:e0000009 ff           ??         FFh
ram:e000000a ff           ??         FFh
ram:e000000b ff           ??         FFh

Decompile: getPart_java.lang.String - (Base64Encoded.class)

1
2 /* Flags:
3     ACC_PUBLIC
4     ACC_STATIC
5
6   public static String getPart()  */
7
8 undefined getPart_java.lang.String(void)
9
10 {
11   return "RlNVY3Rme2wxaWs=";
12 }
13
```

Now moving towards PlainText file which consists of a string “531lfr3f1_” that I just reversed as it was given in FlagChecker. And after reversing I got “_1f3rfl135”



Now we have the last file that is IntEncoded that has 0 to 9 var values that consist of hex values so after decoding them:



Here's the ASCII representation of each hexadecimal value:

- 0x5f = '_'
- 0x31 = '1'
- 0x6e = 'n'
- 0x5f = '_'
- 0x34 = '4'
- 0x5f = '_'
- 0x6a = 'j'
- 0x34 = '4'
- 0x72 = 'r'
- 0x7d = '}'

Concatenating these characters together, we get the string output: "_1n_4_j4r}".

So together we get the flag: **FSUctf{l1ik3_c4tch1ng_1f3rfl135_1n_4_j4r}**

Q2.

In this problem I went through the main function where I saw operation1 and operation2.

Operation1 function consists of revering the string.

Operation2 does shuffling the characters based on the input given to the argument. So I wrote a program that will take output as input and give the original input.

So I wrote a C program that gave me the string : “}5gn1rts_f0_lluf_d3kc4p{ftcusf” when I gave the input as “f5uc1fts4f0_l_ufld3kc_p{rtngs}” that was provided along with the problem.

So then I used cyberchef to reverse the string and that is how I got the flag.

fsuctf{p4ck3d_full_0f_str1ng5}

Decompile: main - (a1.out)

```c

undefined8 main(void)

{
  long in_FS_OFFSET;
  undefined8 local_38;
  undefined8 local_30;
  undefined8 local_28;
  undefined8 local_20;
  undefined2 local_18;
  undefined local_16;
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  local_38 = 0x727b667463757366;
  local_30 = 0x5f64657463616465;
  local_28 = 0x6465746361646572;
  local_20 = 0x657463616465725f;
  local_18 = 0x7d64;
  local_16 = 0;
  Operation1(&local_38);
  Operation2(&local_38);
  puts((char *)&local_38);
  if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
  }
  return 0;
```

Decompile: Operation1 - (a1.out)

```c

void Operation1(char *param_1)

{
  char cVar1;
  size_t sVar2;
  int local_14;
  int local_10;

  sVar2 = strlen(param_1);
  local_10 = (int)sVar2;
  for (local_14 = 0; local_10 = local_10 + -1, local_14 < local_10; local_14 = local_14 + 1) {
    cVar1 = param_1[local_14];
    param_1[local_14] = param_1[local_10];
    param_1[local_10] = cVar1;
  }
  return;
}
```

Decompile: Operation2 - (a1.out)

```
void Operation2(char *param_1)

{
  char cVar1;
  int iVar2;
  size_t sVar3;
  long in_FS_OFFSET;
  int local_40;
  int local_38 [10];
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  sVar3 = strlen(param_1);
  iVar2 = (int)sVar3;
  local_38[0] = 0;
  local_38[1] = 1;
  local_38[2] = 1;
  local_38[3] = 2;
  local_38[4] = 3;
  local_38[5] = 5;
  local_38[6] = 8;
  local_38[7] = 0xd;
  local_38[8] = 0x15;
  for (local_40 = 0; (local_40 < 10 && (local_38[local_40] < iVar2 - local_38[local_40]));
      local_40 = local_40 + 1) {
    cVar1 = param_1[local_38[local_40]];
    param_1[local_38[local_40]] = param_1[(long)(iVar2 - local_38[local_40]) + -1];
```

Run Debug Stop Share Save { } Beautify

main.c

```c
void OP2_reverse(char *input) {
    size_t length = strlen(input);
    int positions[] = {0, 1, 1, 2, 3, 5, 8, 13, 21};
    for (int i = 0; i < sizeof(positions) / sizeof(positions[0]); i++) {
        int pos = positions[i];
        if (pos >= length - pos) {
            break;
        }
        char temp = input[pos];
        input[pos] = input[length - pos - 1];
        input[length - pos - 1] = temp;
    }
```

input

```
Output before reversal: f5uc1fts4f0_l_ufld3kc_p{rtngs}
Input after reversal: }5gn1rts_f0_lluf_d3kc4p{ftcusf


...Program finished with exit code 0
Press ENTER to exit console.
```

Recipe

Reverse

By
Character

Input

```
}5gn1rts_f0_lluf_d3kc4p{ftcusf
```

abc 30 1

Output

```
fsuctf{p4ck3d_full_0f_str1ng5}
```

Q3.

In this problem I first installed angr. I created a python environment and then created the following angr script.

```
import angr
import sys
binary_path = "./challenge"
def load_project(binary_path):
    return angr.Project(binary_path)
def create_initial_state(project):
    return project.factory.entry_state(
        add_options={
            angr.options.SYMBOL_FILL_UNCONSTRAINED_MEMORY,
            angr.options.SYMBOL_FILL_UNCONSTRAINED_REGISTERS
        }
    )
def run_simulation(project, initial_state):
    simulation = project.factory.simgr(initial_state)
    return simulation
def is_successful_execution(state):
    stdout_output = state.posix.dumps(sys.stdout.fileno())
    return b'Correct!' in stdout_output
def should_abort_execution(state):
    stdout_output = state.posix.dumps(sys.stdout.fileno())
    return b'Try again.' in stdout_output
def find_solution(simulation):
    simulation.explore(find=is_successful_execution, avoid=should_abort_execution)
    return simulation.found[0] if simulation.found else None
def main(argv):
    binary_path = "./challenge"
    project = load_project(binary_path)
    initial_state = create_initial_state(project)
    simulation = run_simulation(project, initial_state)
    solution_state = find_solution(simulation)
    if solution_state:
        solution_input = solution_state.posix.dumps(sys.stdin.fileno()).decode()
        print("Solution Input:", solution_input)
    else:
        raise Exception('Could not find the solution')
if __name__ == '__main__':
```

```
main(sys.argv)
```

This Python script utilizes the Angr framework to perform symbolic execution on a binary file named "challenge" located in the current directory. Here's a breakdown of the script:

1. **Imports**: The script imports the necessary modules `angr` and `sys`.
2. **Function Definitions**:
   - load_project(binary_path): This function loads the binary file specified by the binary_path parameter using the angr.Project constructor and returns the project object.
   - create_initial_state(project): This function creates the initial state for symbolic execution. It sets options for symbolic execution such as SYMBOL_FILL_UNCONSTRAINED_MEMORY and SYMBOL_FILL_UNCONSTRAINED_REGISTERS.
   - run_simulation(project, initial_state): This function creates a simulation manager for the project with the initial state and returns it.
   - is_successful_execution(state): This function checks if the state represents a successful execution by checking if the output contains the string "Correct!".
   - should_abort_execution(state): This function checks if the state represents an aborted execution by checking if the output contains the string "Try again.".
   - find_solution(simulation): This function explores the simulation and finds a state that represents a successful execution (is_successful_execution). It returns the first found solution state or None if no solution is found.
3. **Main Function** (main(argv)):
   - It sets the binary_path variable to the path of the binary file.
   - Loads the project using load_project.
   - Creates the initial state using create_initial_state.
   - Runs the simulation using run_simulation.
   - Finds the solution using find_solution.
   - If a solution is found, it prints the solution input. Otherwise, it raises an exception indicating that no solution was found.
4. **Script Execution**: The script checks if it's being run as the main program (__name__ == '__main__') and then calls the main function, passing sys.argv as an argument.

And then I ran this python file and got the flag: **FSUctf{why_u_50_4ngry_b01_0r_g1rl}**

```
(angr_env)kali@kali: ~
File  Actions  Edit  View  Help
Processing triggers for initramfs-tools (0.142) ...
update-initramfs: Generating /boot/initrd.img-6.6.9-amd64
Processing triggers for ca-certificates-java (20240118) ...
done.

┌──(kali㉿kali)-[~]
└─$ sudo apt install virtualenvwrapper
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer required:
  cython3 debtags kali-debtags libadwaita-1-0 libappstream5 libatk-adaptor libboost-dev libboost1.74-dev libhiredis0.14
  libjavascriptcoregtk-4.0-18 libopenblas-dev libopenblas-pthread-dev libopenblas0 libperl5.36 libpython3-all-dev
  libpython3.12 libpython3.12-dev libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediagsttools5
  libqt5multimediawidgets5 librtlsdr0 libstemmer0d libucl1 libwebkit2gtk-4.0-37 libxmlb2 libxsimd-dev libzxing2
  perl-modules-5.36 python3-all-dev python3-backcall python3-beniget python3-debian python3-future python3-gast
  python3-pickleshare python3-pyatspi python3-pythran python3-requests-toolbelt python3-rfc3986 python3-unicodecsv
  python3.12-dev xtl-dev zenity zenity-common
Use 'sudo apt autoremove' to remove them.
```

```
┌──(kali㉿kali)-[~]
└─$ virtualenv angr_env

created virtual environment CPython3.11.8.final.0-64 in 918ms
  creator CPython3Posix(dest=/home/kali/angr_env, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/kali/.lo
hare/virtualenv)
    added seed packages: pip==24.0, setuptools==68.1.2, wheel==0.43.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

┌──(kali㉿kali)-[~]
└─$ ls
angr_env  Desktop  Documents  Downloads  Music  P  Pictures  Public  Templates  Videos
```

```
(angr_env)kali@kali: ~
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/angr_env]
└─$ source angr_env/bin/activate

source: no such file or directory: angr_env/bin/activate

┌──(kali㉿kali)-[~/angr_env]
└─$ cd ..

┌──(kali㉿kali)-[~]
└─$ source angr_env/bin/activate


┌──(angr_env)─(kali㉿kali)-[~]
└─$ ls
angr_env  Desktop  Documents  Downloads  Music  P  Pictures  Public  Templates  Videos

┌──(angr_env)─(kali㉿kali)-[~]
└─$ pip install angr

Collecting angr
  Downloading angr-9.2.95-py3-none-manylinux2014_x86_64.whl.metadata (4.8 kB)
Collecting CppHeaderParser (from angr)
  Downloading CppHeaderParser-2.7.4.tar.gz (54 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.4/54.4 kB 831.0 kB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting GitPython (from angr)
  Downloading GitPython-3.1.42-py3-none-any.whl.metadata (12 kB)
Collecting ailment==9.2.95 (from angr)
  Downloading ailment-9.2.95-py3-none-any.whl.metadata (1.6 kB)
Collecting archinfo==9.2.95 (from angr)
  Downloading archinfo-9.2.95-py3-none-any.whl.metadata (1.9 kB)
Collecting cachetools (from angr)
```

```
(angr_env)kali@kali: ~
File  Actions  Edit  View  Help
Traceback (most recent call last):
  File "/home/kali/angrP.py", line 45, in <module>
    main(sys.argv)
  File "/home/kali/angrP.py", line 33, in main
    project = load_project(binary_path)
              ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/kali/angrP.py", line 5, in load_project
    return angr.Project(binary_path)
           ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/kali/angr_env/lib/python3.11/site-packages/angr/project.py", line 142, in __init__
    raise Exception("Not a valid binary file: %s" % repr(thing))
Exception: Not a valid binary file: './challenge'

┌──(angr_env)─(kali㉿kali)-[~]
└─$ ls
angr_env  angrP.py  Desktop  Documents  Downloads  Music  P  Pictures  Public  Templates  Videos

┌──(angr_env)─(kali㉿kali)-[~]
└─$ cd P

┌──(angr_env)─(kali㉿kali)-[~/P]
└─$ ls
file3.py  file.py  roplon  roplon.c  vuln  vuln.c

┌──(angr_env)─(kali㉿kali)-[~/P]
└─$ cd ..

┌──(angr_env)─(kali㉿kali)-[~]
└─$ python3 angrP.py
Solution Input: FSUctf{why_u_50_4ngry_b01_0r_g1rl}

┌──(angr_env)─(kali㉿kali)-[~]
└─$
```