**PIT TIME DOCUMENTATION**

A screenshot of a computer

AI-generated content may be incorrect.**01 | Problem**

Here we can see the challenge is among Binary Exploitation Category. The Description says “Can you try to get the flag? Beware we have PIE! Additional details will be available after launching your challenge instance.”

**Analysis**

A screenshot of a phone

AI-generated content may be incorrect.Here the hint says about the address found locally and in the server output.

As we run the instance two more file shows up which can be downloaded locally.

A screenshot of a computer

AI-generated content may be incorrect.

First lets run the server.

|  |
| --- |
| Note:  I am using my custom python script to connect to netacat but you can use your linux cmd prompt to connect to the server. |

A screen shot of a computer

AI-generated content may be incorrect.

Here we can see the server gives us an information about the Address of main which is 0x5fc3aed2233d and the prompts us to jump to an address. Lets try to connect to the server again to see if the address changes or not.

A screen shot of a computer program

AI-generated content may be incorrect.

As we can see the addresses have changed again and the address we found seems to be of hex value.

Now let’s try to run the downloaded code locally.

A black background with colorful text

AI-generated content may be incorrect.

Here we can see that even after running the code multiple times the address of main stays the same.

**Solution**

We know that when a C program executes, every function of it is executed in a specific memory address. To see the addresses, we need to enter the debug mode of C where we can inspect in depth how the functions are being executed, and which memory address is being used.

So having that in mind, there is a high possibility the main address found in the server is the memory address of the main function when being executed.

A screenshot of a computer

AI-generated content may be incorrect.

We are given these two files in the download options. So first we will compile the vuln.c using the gcc by the command “gcc vuln.c -o vuln”.

A computer screen with white text

AI-generated content may be incorrect.

After compiling we will get vuln.exe file which can be opened and examined by binary ninja.

A screenshot of a computer

AI-generated content may be incorrect.

First we will select the “Go to Address…” option from the “Edit” Menu.

A screenshot of a computer

AI-generated content may be incorrect.

The we will put the hex value we got when we ran the code locally.

A screenshot of a computer

AI-generated content may be incorrect.

Take note that the memory address of main() function is indeed “0040150f”.

After reviewing the code locally downloaded, we can see that the upon executing the win function we can get the flag.

A screen shot of a computer program

AI-generated content may be incorrect.

Lets talk about what is PIE (Position Independent Executable) is a term to describe when a function can be executed from anywhere in the memory address. The memory address changes every time means it’s random but the distance between the memory address of each functions stays the same.

Now what if we somehow trigger the “win()” function on the server? If the memory addresses of the main functions keep changing but the difference between the win() and main() functions stays the same then we can potentially calculate the difference and trigger the win() function in the server to get the function.

Lets take a look at win() function which can be found after scrolling up a little bit in binary Ninja.

A screenshot of a computer

AI-generated content may be incorrect.

We can see that the memory address of win() function is “0040147e”. Lets subtract the memory address of main() from the memory address of win().

A screenshot of a calculator

AI-generated content may be incorrect.

We can see that the difference is 0x91.

Now let’s run the program on the server to test if our method works or not.

A screen shot of a computer

AI-generated content may be incorrect.

And as we can see we got the flag after subtracting 0x555de3bf633d from 0x91 which is 0x555de3bf62ac.