snake

Category: Ghidra (0xB)
Points: 150

Description

> This is a hard challenge to reverse. Can you figure it out?
> `nc vunrotc.cole-ellis.com 11300`

The goal of this challenge is to understand how Ghidra works when binaries are not written directly in C, and use an alternative route to decompile the binary.

Flag

```
flag{written_in_python_rev_in_c}
```

Solution

This isn't the most important of challenges but gives insight into how Ghidra isn't perfect. It also shows what happens if a binary is *stripped*.

BLUF: **This challenge was not written in C**. This challenge was designed in Python and converted into an executable.

If you run `file` on the binary, you notice the binary is *stripped*. This means all function names and symbols are removed from the binary. You'll notice this when you open Ghidra; all the function names are `FUN_<addr>` based on their location `addr` in memory.

We search for a Python decompiler and find pyinstxtractor. This extracts Python bytecode files from the executable. Then, we need to extract the Python code from the bytecode. We can do this using uncompyle6 or decompyle3.

Once this is done, we'll have source code with an `encode()` function. Reversing this on the output prints out the flag.