Rev
Aplet321
Author: kaiphait

1 Description

This program expects to recieve in input some sentences in order to convince the algorith to give the flag.
Flag: lactf{next_year_i'll_make_aplet456_hqp3c1a7bip5bmnc}

2 Scoreboard Description

Unlike Aplet123, Aplet321 might give you the flag if you beg him enough.

3 Solution

We need to decompile the ELF file in order to understand the logic of the program. There is only the main function, and as we scroll through the code, we can see that the program is expecting to receive specific words in order to give us the flag.

```
for (int i = 0; i + 6 <= input_len; ++i) {
    if (strncmp(input + i, "pretty", 6) == 0) {
        ++pretty_cnt;
    }
    if (strncmp(input + i, "please", 6) == 0) {
        ++please_cnt;
    }
}
```

Scrolling more in the code we notice that the algorithm is expecting a specific number of the words

```
if (please_cnt == 0) {
    puts("so rude");
} else if (strstr(input, "flag")) {
    if (pretty_cnt + please_cnt == 54 && pretty_cnt - please_cnt == -24) {
        puts("ok here's your flag");
        system("cat flag.txt");
    } else {
        puts("sorry, i'm not allowed to do that");
    }
} else {
    puts("sorry, i didn't understand what you mean");
}
```

it is expecting exactly 54 words as the sum of the words "pretty" and "please", and exactly -24 as the difference between the two words. This means that "pretty" must be written 15 times and "please" must be written 39 times.

known this we can start create our `solve.py` script:

```
#!/usr/bin/env python3

from pwn import *

exe = ELF("./aplet321")

context.binary = exe

if args.REMOTE:
    r = remote("chall.lac.tf", 31321)
else:
    r = process([exe.path])
    if args.GDB:
        gdb.attach(r)

r.sendline(b'pretty ' * 15 + b'please ' * 39 + b'flag')

r.interactive()
```