COMPSCI 561: CTF Solution

Will Lillis

April 24, 2023

The encrypted flag (as given in `flag.enc`) is “`TA5YRwxHQQ4GAwkAVgYH`”. The decrypted flag to be ”captured” is “`thisistheflag00`”. While there are many ways to go about solving this CTF, I intended for this process to proceed as follows.

- The user runs the command `strings flag` (as hinted in the instructions PDF) and discovers two hints left in the program binary. These hints are:
  - “HINT1: The encryption algorithm works by performing a bitwise XOR,” ”character by character, with the key buffer. The result of this is then encoded using base64 encoding.”
  - “HINT2: Value of 0 passed for argument ’key’. Using default key value MD5(7) to construct key buffer”
- With these hints, the user can then figure out the encryption algorithm and deduce that MD5 hash of 7, “`8f14e45fceea167a5a36dedd4bea2543`”, was used as the key buffer.
- Because XOR is a reversible operation, the user could then write some code (either using `decrypt.c` or something else) to decrypt the flag. Some sample code that accomplishes this is included in `decrypt_solved.c` (in this repo, but not the container). The decryption portion of this code is:

```
size_t size_out;
char* decoded = base64_decode(user_in, strlen(user_in), &size_out);
const char* key_buff = "8f14e45fceea167a5a36dedd4bea2543"; // MD5(7)
for (int i = 0; i < size_out; i++) {
        decoded[i] ^= key_buff[i];
}
printf("decrypted: %s\n", decoded);
```

Running `decrypt.c` with the above code pasted at the end (and inputting the encrypted flag when prompted) will print the decrypted flag to the console.