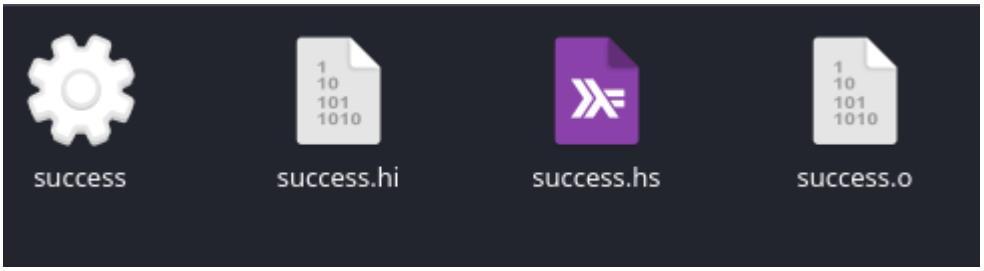
Success

Category: Reverse Engineering

Provided Source file (Initially success.hs was only given):



Initial Observation

We are given a haskell source code `success.hs`.

Looking at the source code, it's very messy:

```
import System.Environment (getArgs)
import System.Exit (exitFailure, exitSuccess)
import Data.Bits

checkFlag :: String -> IO ()
checkFlag flag = do
   if length flag /= 39
       then do
           putStrLn "bruh its 39 characters long"
           exitFailure
       else do
           let chars :: [Integer]
               chars = map fromIntegral . map fromEnum $ flag
           if (*) (chars !! 37) (chars !! 15) == 3366 then do
               if (+) (chars !! 8) (chars !! 21) == 197 then do
                   if (*) (chars !! 8) (chars !! 13) == 9215 then do
                       if (*) (chars !! 0) (chars !! 3) == 2714 then do
                           if (+) (chars !! 3) (chars !! 21) == 159 then
do
                               if (*) (chars !! 1) (chars !! 20) == 5723
then do
                                   if (.|.) (chars !! 6) (chars !! 37) ==
105 then do
                                       if (*) (chars !! 11) (chars !! 7)
```

```
== 11990 then do
                                                  if (.&.) (chars !! 29) (chars
!! 25) == 100 then do
                                                      if (.|.) (chars !! 16)
(chars !! 29) == 127 then do
                                                          if (-) (chars !! 20)
(chars !! 6) == -8 then do
                                                              if (+) (chars !!
21) (chars !! 20) == 197 then do
                                                                  if (+) (chars
!! 2) (chars !! 36) == 77 then do
                                                                      if (*)
(chars !! 35) (chars !! 11) == 3630 then do
                                                                          if (*)
(chars !! 4) (chars !! 3) == 2714 then do
                                                                              if
xor (chars !! 35) (chars !! 6) == 72 then do
if (+) (chars !! 25) (chars !! 24) == 221 then do
if (*) (chars !! 14) (chars !! 36) == 3465 then do
if (-) ((-) (chars !! 15) (chars !! 11)) 148 == -156 then do
if (+) (chars !! 37) (chars !! 17) == 138 then do
if xor (chars !! 1) (chars !! 38) == 70 then do
if (+) (chars !! 9) (chars !! 29) == 212 then do
if (-) (chars !! 30) (chars !! 10) == 7 then do
if (+) (chars !! 10) (chars !! 33) == 206 then do
if (*) (chars !! 7) (chars !! 15) == 11118 then do
if (*) ((*) (chars !! 28) (chars !! 14)) 55 == 641025
then do
if (.|.) ((.|.) (chars !! 7) (chars !! 4)) 216 ==
255 then do
if (+) (chars !! 24) (chars !! 4) == 151 then
do
if (*) (chars !! 2) (chars !! 30) == 4928
then do
if (+) (chars !! 5) (chars !! 22) == 2
24 then do
```

```
if (.|.) (chars !! 18) (chars !! 3
6) == 127 then do

if (+) (chars !! 13) (chars !!
7) == 195 then do

if (.|.) (chars !! 9) (cha
rs !! 17) == 111 then do

if (*) (chars !! 12) (
chars !! 9) == 10403 then do

if xor (chars !! 2
8) (chars !! 27) == 23 then do

if xor (chars
!! 13) (chars !! 34) == 59 then do

if (+) (ch
ars !! 18) (chars !! 31) == 200 then do

if (+)
(chars !! 17) (chars !! 32) == 213 then do

if
(*) (chars !! 2) (chars !! 12) == 4444 then do

 if (*) (chars !! 24) (chars !! 31) == 11025 then do

     if (*) (chars !! 5) (chars !! 0) == 5658 then do

         if (+) ((+) (chars !! 10) (chars !! 32)) 228 == 441 then do

             if (*) (chars !! 35) (chars !! 0) == 1518 then do

                 if (.|.) (chars !! 30) (chars !! 8) == 113 then do

                     if (-) (chars !! 28) (chars !! 34) == 11 then do

                         if (*) (chars !! 26) (chars !! 14) == 9975 then
do

                             if (*) (chars !! 31) (chars !! 22) == 10605
then do

                                 if (*) ((*) (chars !! 26) (chars !! 32))
239 == 2452140 then do

                                     if (*) (chars !! 28) (chars !! 38) ==
```

```
13875 then do

                                        if (+) (chars !! 18) (chars !!
16) == 190 then do

                                            if (+) ((+) (chars !! 27)
(chars !! 26)) 96 == 290 then do

                                                if (-) (chars !! 22)
(chars !! 38) == -24 then do

                                                    if (+) (chars !! 33)
(chars !! 5) == 224 then do

                                                        if (*) (chars !!
19) (chars !! 16) == 10355 then do

                                                            if (+) (chars
!! 27) (chars !! 1) == 158 then do

                                                                if (+)
(chars !! 33) (chars !! 12) == 202 then do

                                                                    if
(*) (chars !! 19) (chars !! 23) == 10355 then do

putStrLn "yea go submit the flag"

exitSuccess

                                                                    else
do

putStrLn "no thats not 10355"

exitFailure

                                                                else do


putStrLn "no thats not 202"

exitFailure

                                                            else do
```

```
                                                                putStrLn
"no thats not 158"

exitFailure

                                                            else do

                                                                putStrLn "no
thats not 10355"

                                                                exitFailure

                                                        else do

                                                            putStrLn "no
thats not 224"

                                                            exitFailure

                                                    else do

                                                        putStrLn "no thats
not -24"

                                                        exitFailure

                                                else do

                                                    putStrLn "no thats not
290"

                                                    exitFailure

                                            else do

                                                putStrLn "no thats not 190"

                                                exitFailure

                                        else do

                                            putStrLn "no thats not 13875"

                                            exitFailure

                                    else do

                                        putStrLn "no thats not 2452140"
```

```
                                    exitFailure
                            else do
                                putStrLn "no thats not 10605"
                                exitFailure
                        else do
                            putStrLn "no thats not 9975"
                            exitFailure
                    else do
                        putStrLn "no thats not 11"
                        exitFailure
                else do
                    putStrLn "no thats not 113"
                    exitFailure
            else do
                putStrLn "no thats not 1518"
                exitFailure
        else do
            putStrLn "no thats not 441"
            exitFailure
    else do
        putStrLn "no thats not 5658"
        exitFailure
else do
    putStrLn "no thats not 11025"
    exitFailure
```

```
el
se do

 putStrLn "no thats not 4444"

 exitFailure

else d
o

pu
tStrLn "no thats not 213"

ex
itFailure

else do

putStr
Ln "no thats not 200"

exitFa
ilure

else do

putStrLn "
no thats not 59"

exitFailur
e

else do

putStrLn "no t
hats not 23"

exitFailure

else do

putStrLn "no thats
not 10403"

exitFailure

else do

putStrLn "no thats not
```

```
111"

exitFailure

else do

putStrLn "no thats not 195
"

exitFailure

else do

putStrLn "no thats not 127"

exitFailure

else do

putStrLn "no thats not 224"

exitFailure

else do

putStrLn "no thats not 4928"

exitFailure

else do

putStrLn "no thats not 151"

exitFailure

else do

putStrLn "no thats not 255"

exitFailure

else do

putStrLn "no thats not 641025"

exitFailure

else do

putStrLn "no thats not 11118"
```

```
exitFailure

else do

putStrLn "no thats not 206"

exitFailure

else do

putStrLn "no thats not 7"

exitFailure

else do

putStrLn "no thats not 212"

exitFailure

else do

putStrLn "no thats not 70"

exitFailure

else do

putStrLn "no thats not 138"

exitFailure

else do

putStrLn "no thats not -156"

exitFailure

else do

putStrLn "no thats not 3465"

exitFailure

else do

putStrLn "no thats not 221"

exitFailure
```

```
else do

putStrLn "no thats not 72"

exitFailure
                                                                  else do

putStrLn "no thats not 2714"

exitFailure
                                                                      else do

putStrLn "no thats not 3630"

exitFailure
                                                                          else do
                                                                              putStrLn
"no thats not 77"
                                                                              exitFailure
                                                                      else do
                                                                          putStrLn "no
thats not 197"
                                                                          exitFailure
                                                                  else do
                                                                      putStrLn "no thats
not -8"
                                                                      exitFailure
                                                              else do
                                                                  putStrLn "no thats not
127"
                                                                  exitFailure
                                                          else do
                                                              putStrLn "no thats not 100"
                                                              exitFailure
                                                      else do
                                                          putStrLn "no thats not 11990"
                                                          exitFailure
                                                  else do
                                                      putStrLn "no thats not 105"
                                                      exitFailure
                                              else do
                                                  putStrLn "no thats not 5723"
                                                  exitFailure
                                          else do
                                              putStrLn "no thats not 159"
                                              exitFailure
                                  else do
                                      putStrLn "no thats not 2714"
                                      exitFailure
```

```
                    else do
                        putStrLn "no thats not 9215"
                        exitFailure
                else do
                    putStrLn "no thats not 197"
                    exitFailure
            else do
                putStrLn "no thats not 3366"
                exitFailure


main :: IO ()
main = do
   args <- getArgs
   case args of
       [flag] -> checkFlag flag
       _      -> do
           putStrLn "yo u gotta pass in a flag bruh or else the case args
thing won't match the first case"
```

After staring at the screen for 4 hours, I realized that i just need the relevant part of the code,

The Relevant part:

```
let chars :: [Integer]
               chars = map fromIntegral . map fromEnum $ flag
           if (*) (chars !! 37) (chars !! 15) == 3366 then do
               if (+) (chars !! 8) (chars !! 21) == 197 then do
                   if (*) (chars !! 8) (chars !! 13) == 9215 then do
                       if (*) (chars !! 0) (chars !! 3) == 2714 then do
                           if (+) (chars !! 3) (chars !! 21) == 159 then
do
                               if (*) (chars !! 1) (chars !! 20) == 5723
then do
                                   if (.|.) (chars !! 6) (chars !! 37) ==
105 then do
                                       if (*) (chars !! 11) (chars !! 7)
== 11990 then do
                                           if (.&.) (chars !! 29) (chars
!! 25) == 100 then do
                                               if (.|.) (chars !! 16)
(chars !! 29) == 127 then do
                                                   if (-) (chars !! 20)
(chars !! 6) == -8 then do
                                                       if (+) (chars !!
21) (chars !! 20) == 197 then do
                                                           if (+) (chars
!! 2) (chars !! 36) == 77 then do
                                                               if (*)
```

```
(chars !! 35) (chars !! 11) == 3630 then do
                                                                    if (*)
(chars !! 4) (chars !! 3) == 2714 then do
                                                                        if
xor (chars !! 35) (chars !! 6) == 72 then do

if (+) (chars !! 25) (chars !! 24) == 221 then do

if (*) (chars !! 14) (chars !! 36) == 3465 then do

if (-) ((-) (chars !! 15) (chars !! 11)) 148 == -156 then do

if (+) (chars !! 37) (chars !! 17) == 138 then do

if xor (chars !! 1) (chars !! 38) == 70 then do

if (+) (chars !! 9) (chars !! 29) == 212 then do

if (-) (chars !! 30) (chars !! 10) == 7 then do

if (+) (chars !! 10) (chars !! 33) == 206 then do

if (*) (chars !! 7) (chars !! 15) == 11118 then do

if (*) ((*) (chars !! 28) (chars !! 14)) 55 == 641025
then do

if (.|.) ((.|.) (chars !! 7) (chars !! 4)) 216 ==
255 then do

if (+) (chars !! 24) (chars !! 4) == 151 then
do

if (*) (chars !! 2) (chars !! 30) == 4928
then do

if (+) (chars !! 5) (chars !! 22) == 2
24 then do

if (.|.) (chars !! 18) (chars !! 3
6) == 127 then do

if (+) (chars !! 13) (chars !!
7) == 195 then do

if (.|.) (chars !! 9) (cha
rs !! 17) == 111 then do

if (*) (chars !! 12) (
chars !! 9) == 10403 then do
```

```
if xor (chars !! 2
8) (chars !! 27) == 23 then do

if xor (chars
!! 13) (chars !! 34) == 59 then do

if (+) (ch
ars !! 18) (chars !! 31) == 200 then do

if (+)
(chars !! 17) (chars !! 32) == 213 then do

if
(*) (chars !! 2) (chars !! 12) == 4444 then do

 if (*) (chars !! 24) (chars !! 31) == 11025 then do

     if (*) (chars !! 5) (chars !! 0) == 5658 then do

         if (+) ((+) (chars !! 10) (chars !! 32)) 228 == 441 then do

             if (*) (chars !! 35) (chars !! 0) == 1518 then do

                 if (.|.) (chars !! 30) (chars !! 8) == 113 then do

                     if (-) (chars !! 28) (chars !! 34) == 11 then do

                         if (*) (chars !! 26) (chars !! 14) == 9975 then
do

                             if (*) (chars !! 31) (chars !! 22) == 10605
then do

                                 if (*) ((*) (chars !! 26) (chars !! 32))
239 == 2452140 then do

                                     if (*) (chars !! 28) (chars !! 38) ==
13875 then do

                                         if (+) (chars !! 18) (chars !!
16) == 190 then do

                                             if (+) ((+) (chars !! 27)
(chars !! 26)) 96 == 290 then do

                                                 if (-) (chars !! 22)
(chars !! 38) == -24 then do

                                                     if (+) (chars !! 33)
```

```
(chars !! 5) == 224 then do

                                                    if (*) (chars !!
19) (chars !! 16) == 10355 then do

                                                        if (+) (chars
!! 27) (chars !! 1) == 158 then do

                                                            if (+)
(chars !! 33) (chars !! 12) == 202 then do

                                                                if
(*) (chars !! 19) (chars !! 23) == 10355 then do


putStrLn "yea go submit the flag"
```

We only need the conditional if-else-if ladder above (disaster), to construct the flag.

Here is the script i used:

```
from z3 import *

s = Solver()

# Declare 39 character variables, each an 8-bit integer (ASCII)
chars = [BitVec(f'c{i}', 8) for i in range(39)]

# All characters should be printable ASCII
for c in chars:
   s.add(c >= 32, c <= 126)

# Constraints (fixed bitwise expressions)
s.add(chars[37] * chars[15] == 3366)
s.add(chars[8] + chars[21] == 197)
s.add(chars[8] * chars[13] == 9215)
s.add(chars[0] * chars[3] == 2714)
s.add(chars[3] + chars[21] == 159)
s.add(chars[1] * chars[20] == 5723)
s.add((chars[6] | chars[37]) == 105)
s.add(chars[11] * chars[7] == 11990)
s.add((chars[29] & chars[25]) == 100)
s.add((chars[16] | chars[29]) == 127)
s.add(chars[20] - chars[6] == -8)
s.add(chars[21] + chars[20] == 197)
s.add(chars[2] + chars[36] == 77)
s.add(chars[35] * chars[11] == 3630)
s.add(chars[4] * chars[3] == 2714)
s.add((chars[35] ^ chars[6]) == 72)
```

```
s.add(chars[25] + chars[24] == 221)
s.add(chars[14] * chars[36] == 3465)
s.add((chars[15] - chars[11]) - 148 == -156)
s.add(chars[37] + chars[17] == 138)
s.add((chars[1] ^ chars[38]) == 70)
s.add(chars[9] + chars[29] == 212)
s.add(chars[30] - chars[10] == 7)
s.add(chars[10] + chars[33] == 206)
s.add(chars[7] * chars[15] == 11118)
s.add(chars[28] * chars[14] * 55 == 641025)
s.add(((chars[7] | chars[4]) | 216) == 255)
s.add(chars[24] + chars[4] == 151)
s.add(chars[2] * chars[30] == 4928)
s.add(chars[5] + chars[22] == 224)
s.add((chars[18] | chars[36]) == 127)
s.add(chars[13] + chars[34] == 195)
s.add((chars[9] | chars[17]) == 111)
s.add(chars[12] * chars[9] == 10403)
s.add((chars[25] ^ chars[27]) == 23)
s.add((chars[13] ^ chars[34]) == 59)
s.add(chars[18] + chars[31] == 200)
s.add(chars[17] + chars[32] == 213)
s.add(chars[2] * chars[12] == 4444)
s.add(chars[24] * chars[31] == 11025)
s.add(chars[5] * chars[0] == 5658)
s.add(chars[10] + chars[32] + 228 == 441)
s.add(chars[35] * chars[0] == 1518)
s.add((chars[30] | chars[8]) == 113)
s.add(chars[28] - chars[34] == 11)
s.add(chars[26] * chars[14] == 9975)
s.add(chars[31] * chars[22] == 10605)
s.add(chars[26] * chars[32] * 239 == 2452140)
s.add(chars[28] * chars[38] == 13875)
s.add(chars[18] + chars[16] == 190)
s.add(chars[27] + chars[26] + 96 == 290)
s.add(chars[22] - chars[38] == -24)
s.add(chars[33] + chars[5] == 224)
s.add(chars[19] * chars[16] == 10355)
s.add(chars[27] + chars[1] == 158)
s.add(chars[33] + chars[12] == 202)
s.add(chars[19] * chars[23] == 10355)

# Solve
if s.check() == sat:
   m = s.model()
   flag = ''.join([chr(m[c].as_long()) for c in chars])
   print("Flag:", flag)
else:
   print("No solution found.")
```

Output:

```
$ python3 flag.py
Flag: .;,;.{imagine_if_i_made_it_compiled!!!}
```

Initially, we were just given a haskell source code so, to check if the flag works, we must compile the source code:

To compile and run a Haskell file (like your `success.hs`), you have several options. The most common tool is **GHC** (Glasgow Haskell Compiler).

Compiling with GHC

1. **Open a terminal** and navigate to the directory containing your `success.hs` file.
2. **Compile the file** with:

```
$ ghc success.hs
```

This will produce an executable named `success` (or `success.exe` on Windows)

- **Run the executable** with your flag as a command-line argument:

text

```
$ ./success "flag here"
```

Replace `"your_flag_here"` with the flag you want to test.

So testing the flag, we get:

```
./success '.;,;.{imagine_if_i_made_it_compiled!!!}'

yea go submit the flag
```

DONE!

What was the challenge even about?

This is what I was thinking when i was just started out this challenge, i bet you thought it too, while reading this, here is the summary:

Basically, explaining what the script/exploit does will be easier to explain:

The script is a **Z3 constraint solver script** that finds a flag (a 39-character ASCII string) by solving a set of mathematical and bitwise constraints. Here's a breakdown of *why* the flag

`'.;,;.{imagine_if_i_made_it_compiled!!!}'` worked:

What the script does

1. **Defines 39 8-bit character variables** — each `chars[i]` represents one character in the flag.
2. **Constrains them to be printable ASCII** (between `32` and `126` inclusive).
3. **Adds 50+ complex constraints** involving:
   - Additions (e.g. `chars[8] + chars[21] == 197`)
   - Multiplications (e.g. `chars[11] * chars[7] == 11990`)
   - Bitwise operations: `|`, `&`, `^`
   - Mixed equations: expressions combining multiple operations or characters
4. **Uses Z3 to solve all constraints simultaneously**.

The Z3 solver works by:

- Systematically evaluating all the constraints.
- Applying SAT-solving techniques to narrow down values.
- Finding a consistent assignment of values (`model`) such that **every constraint is satisfied**.

The constraints were designed in a way that:

- Has a **unique solution** (or one of very few).
- Is computationally hard by hand — but solvable via symbolic methods like Z3.

In Short:

This challenge is a constraint-based **flag validator** puzzle. It's about **understanding and solving the internal logic** (math/bitwise conditions) the program uses to check the correctness of the input. You didn't reverse it — you need to solve it *symbolically*.