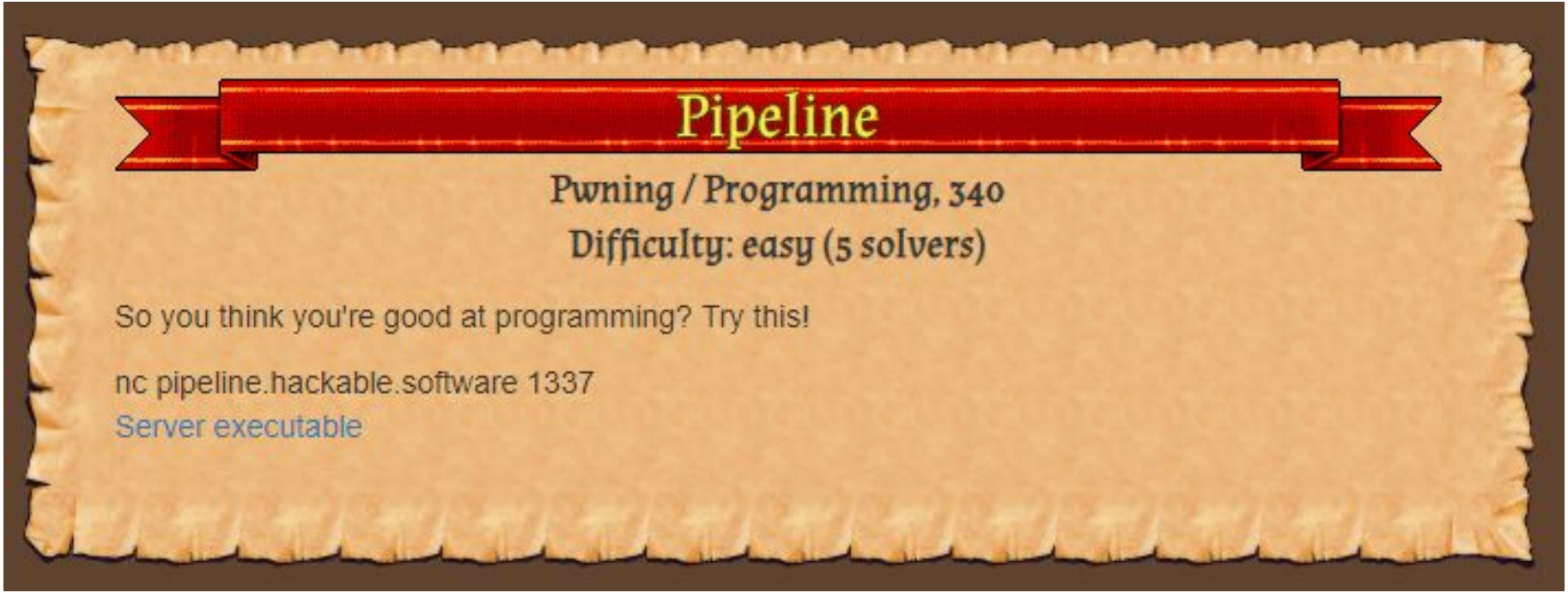
Pipeline - j00ru



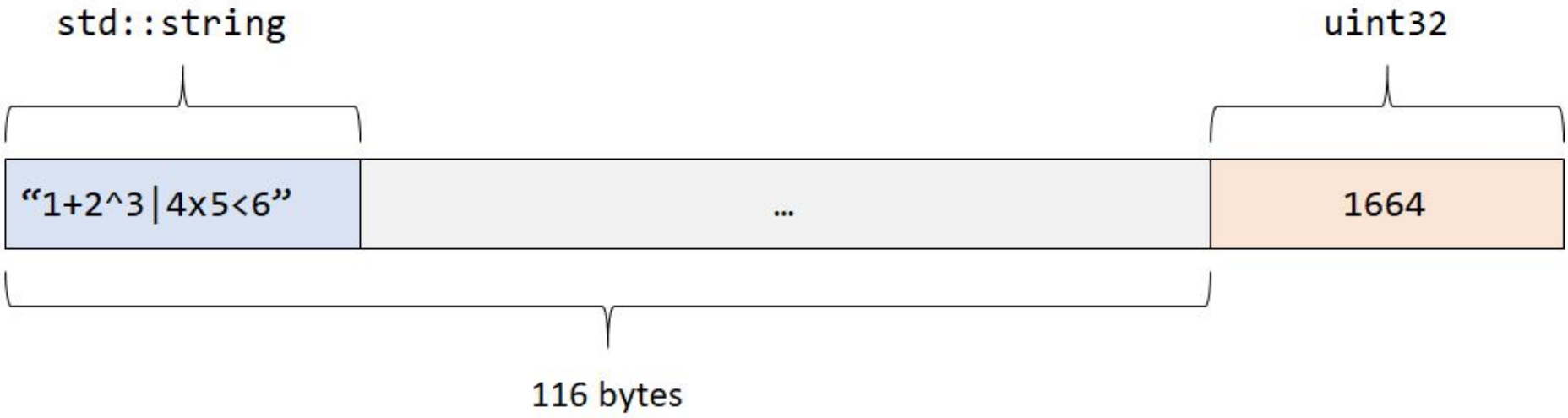
Pipeline - j00ru

- Fully hardened Linux x64 binary
- Accepts a 2D program with maximum dimensions of 80x25
- Upon research, you can find it's the Befunge esoteric language without a few instructions
  - Stack-based machine, potentially self-modifying. Instruction pointer can move in four different directions
- The user-provided program is run against 20 arithmetic / logical expressions consisting of digits and operators (add, multiply, exponent, or, xor, rol, ror)
- For example, 1+2^3|4x5<6 = 1664
- If the program produces valid output (as a text string) for all tests, you win the flag

Pipeline - j00ru

- Hardcore programming task?
  - Exponentiation, or, xor, rol, ror not natively supported in Befunge
  - Very difficult or impossible within the small 80x25 board
- Idea: stack-based out-of-bounds read from the input buffer can be used to read the expected result
- The input data is represented as `std::string`, but because it is always <=16 bytes long, it is *inlined* in the object space (not a separate allocation)
- If you use the ~ operator 116 times, you will reach the correct output stored as uint32.

Pipeline - j00ru



Pipeline - j00ru

- The expression value can be read byte by byte in little endian
- Makes the task much easier, but you still have to:
- Convert the four bytes into a 32-bit integer
- Convert the integer to a string
- Skip the leading zeros

Pipeline - j00ru

```
60
11
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~v
v~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~<
>~~884***+~884**:**+~884**::***+443**01p:55+%01g+90p55+/   v
:55+%01g+80p55+/:55+%01g+70p55+/:55+%01g+60p55+/          v>
:55+%01g+50p55+/:55+%01g+40p55+/:55+%01g+30p55+/         v>
:55+%01g+20p55+/:55+%01g+10p55+/:55+%01g+00p55+/        v>
 0>:0g01g-#v_:9-#v_v                                   v>
  ^            +1<
v          <     < <
>:0g,:9-#v_@
^      +1<
```

Pipeline - j00ru

```
Test  1 / 20: PASSED
Test  2 / 20: PASSED
Test  3 / 20: PASSED
Test  4 / 20: PASSED
Test  5 / 20: PASSED
Test  6 / 20: PASSED
Test  7 / 20: PASSED
Test  8 / 20: PASSED
Test  9 / 20: PASSED
Test 10 / 20: PASSED
Test 11 / 20: PASSED
Test 12 / 20: PASSED
Test 13 / 20: PASSED
Test 14 / 20: PASSED
Test 15 / 20: PASSED
Test 16 / 20: PASSED
Test 17 / 20: PASSED
Test 18 / 20: PASSED
Test 19 / 20: PASSED
Test 20 / 20: PASSED
[+] Congratulations, here's your flag: DrgnS{Es0t3ric_l4nguage_is_th3_be5t_languag3}
```