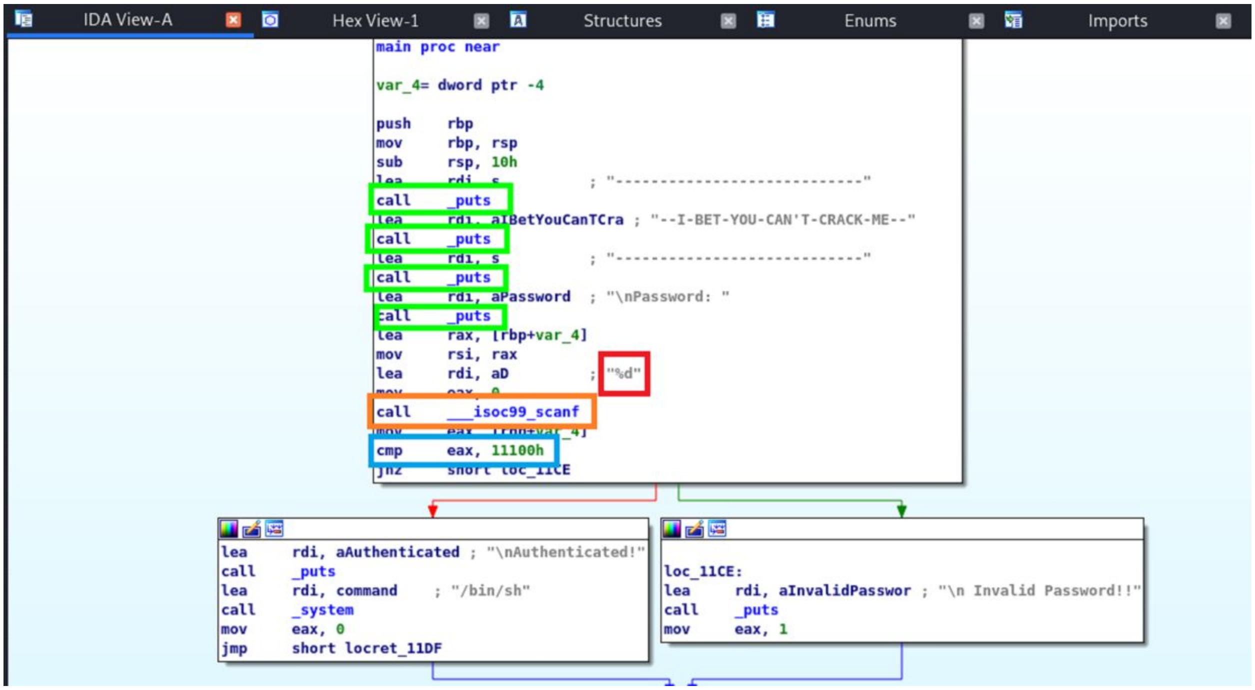
Forgotmypass Reversing Challenge Writeup
4/9/20 10:20 am
Learning Sources, Live Overflow Youtube Course, Used information from his Buffer Overflow Series

Used IDA



```
main proc near

var_4= dword ptr -4

push    rbp
mov     rbp, rsp
sub     rsp, 10h
lea     rdi, s          ; "---------------------------"
call    _puts
lea     rdi, aIBetYouCanTCra ; "--I-BET-YOU-CAN'T-CRACK-ME--"
call    _puts
lea     rdi, s          ; "---------------------------"
call    _puts
lea     rdi, aPassword  ; "\nPassword: "
call    _puts
lea     rax, [rbp+var_4]
mov     rsi, rax
lea     rdi, aD         ; "%d"
mov     eax, 0
call    ___isoc99_scanf
mov     eax, [rbp+var_4]
cmp     eax, 11100h
jnz     short loc_11CE
```

```
lea     rdi, aAuthenticated ; "\nAuthenticated!"
call    _puts
lea     rdi, command    ; "/bin/sh"
call    _system
mov     eax, 0
jmp     short locret_11DF
```

```
loc_11CE:
lea     rdi, aInvalidPasswor ; "\n Invalid Password!!"
call    _puts
mov     eax, 1
```

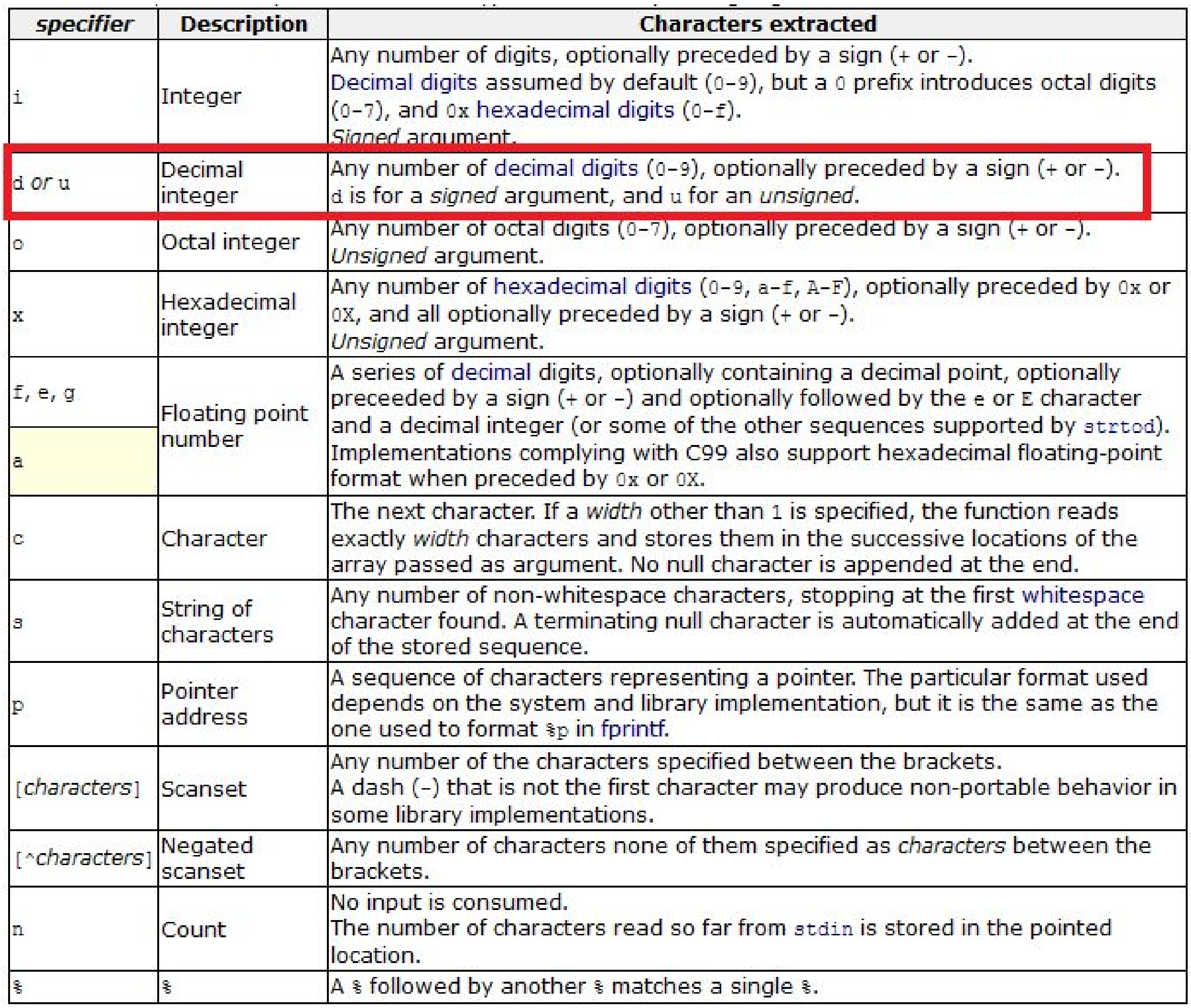
CALL PUTS: GREEN
puts is a function in C which will print text
So this most likely prints the banner “I bet you can't crack me” we see when we open the program

http://www.cplusplus.com/reference/cstdio/puts/

CALL SCANF: ORANGE
scanf is a function which takes user input
the first argument is the format string in which is specified which type of value will be given and the second argument is the variable in which this value should be stored

http://www.cplusplus.com/reference/cstdio/scanf/

RED:
looking at the comments in ida we can see that in this case the %d format specifier is used this is used for a decimal value.



specifier	Description	Characters extracted
i	Integer	Any number of digits, optionally preceded by a sign (+ or -). Decimal digits assumed by default (0-9), but a 0 prefix introduces octal digits (0-7), and 0x hexadecimal digits (0-f). *Signed* argument.
d *or* u	Decimal integer	Any number of decimal digits (0-9), optionally preceded by a sign (+ or -). d is for a *signed* argument, and u for an *unsigned*.
o	Octal integer	Any number of octal digits (0-7), optionally preceded by a sign (+ or -). *Unsigned* argument.
x	Hexadecimal integer	Any number of hexadecimal digits (0-9, a-f, A-F), optionally preceded by 0x or 0X, and all optionally preceded by a sign (+ or -). *Unsigned* argument.
f, e, g	Floating point number	A series of decimal digits, optionally containing a decimal point, optionally preceeded by a sign (+ or -) and optionally followed by the e or E character and a decimal integer (or some of the other sequences supported by strtod). Implementations complying with C99 also support hexadecimal floating-point format when preceded by 0x or 0X.
a		
c	Character	The next character. If a *width* other than 1 is specified, the function reads exactly *width* characters and stores them in the successive locations of the array passed as argument. No null character is appended at the end.
s	String of characters	Any number of non-whitespace characters, stopping at the first whitespace character found. A terminating null character is automatically added at the end of the stored sequence.
p	Pointer address	A sequence of characters representing a pointer. The particular format used depends on the system and library implementation, but it is the same as the one used to format %p in fprintf.
[*characters*]	Scanset	Any number of the characters specified between the brackets. A dash (-) that is not the first character may produce non-portable behavior in some library implementations.
[^*characters*]	Negated scanset	Any number of characters none of them specified as *characters* between the brackets.
n	Count	No input is consumed. The number of characters read so far from stdin is stored in the pointed location.
%	%	A % followed by another % matches a single %.

CMP EAX, 11100h: BLUE
CMP is an instruction which makes a comparison between two values
after this the code flow can go two ways, either the block which contains the success message (left)
or the block which contains the invalid password message
in this case the comparison is done against a set number of 11100h (hexadecimal notation)
when we convert this into decimal we get the number 69888
which turned out to be the password value
https://c9x.me/x86/html/file_module_x86_id_35.html