Forums / Assignments Help

Not understanding approach to solve Question 9

Subscribe for email updates.

UNRESOLVED

project1 × question9 × + Add Tag

Sort replies by: Oldest first Newest first Most popular

Aniket Kamat · 10 days ago

HI guys,
So I am trying the following
I got &buf = 0xbffff130
&p = 0xbffff530 (+1024)
&s = 0xbffff538 (+4 integer)
&tmp = 0xbffff53c(again +4 integer)

I think what i am supposed to do is overwrite the value of tmp() so that the input to s executes directly at pat_on_back i.e 0xbffff530

But i am not understanding how to make the buffer overflow. I have reviewed the pointer arithmetic threads.
Can anyone tell me where do I need to exactly put the input?

↑ 0 ↓ · flag

Aniket Kamat · 10 days ago

Also -10\x34\xf5\xff\xbf is giving me the execution back to pat-on-back without nay error but what does it mean unsigned int?
Please help me clear the concept.
thanks

↑ 0 ↓ · flag

+ Comment

Philip McShane · 10 days ago 🔗 ★ APPROVED

This question wants you to find the int value to enter so that pat_on_back will execute.

Unsigned int means an integer value not proceed by a + or - sign.

If you enter 1 tmp() will execute the function at index 1, if you enter 2 it will execute the function at index 2.
You need to find the int that you enter so that the buffer overflows casing tmp() to execute pat_on_back.
Think of it has trying to find what index p would be at in ptrs..

So if p was the fourth value in ptrs (Index 3) the difference in memory would be 12.

Look at the difference between &ptrs[0] and &p.
Remember that the address' go up by 4 each time

↑ 6 ↓ · flag

Peter · 10 days ago 🔗

I am sorry why &ptrs (0x804a0d4) and &p (0xbffff534) , the 1st is at the heap and the other is on the stack. do you mean &ptrs and p itself or address for pat on the back? and how would I point ptrs to it ?

↑ 0 ↓ · flag

Aniket Kamat · 10 days ago 🔗

Yea I have the same question..i calculated the differences but it does not make sense to me.

↑ 0 ↓ · flag

Philip McShane · 10 days ago 🔗

To understand this it might help to look at how the program works without using exploits.

Lets say the user enters 1.
s is set as 1 and tmp is set as a pointer to ptrs[1].
Notice that the functions get_wisdom and put_wisdom aren't explicitly called in main.
Instead an array of pointers is used to determine what function is called at line 102.
When we reach line 102 we get tmp();
which points to the memory location 0x804a0d8 (the same as &ptrs[1])
The value at 0x804a0d8 (ptrs[1]) is another pointer which points at 0x804857e (the memory location for gets_wisdom)
This causes the get_wisdom function to execute as if line 102 had been gets_wisdom();

What we are trying to do is get the pointer which would point at the ptrs array to point at a pointer for pat_on_back.
In other words we want line 102 to be the value for ptrs[s] which points to &p which then points at pat_on_back.
Just as in the above example where line 102 was the value for ptrs[s] which pointed at a pointer for get_wisdom.

By the way, as p is a pointer to pat_on_back p is the same as the address for pat_on_back.
We want the address of the pointer p not its value.

↑ 10 ↓ · flag

Peter · 10 days ago 🔗

What I thought is that I need to move ptrs till it reaches &p, so basically s=no. of address jumps to &p
and when I do so, either using Hex and little endian or decimal to enter to buf, it still crushes the program
what I am calculating wrong ? &p - &ptrs

↑ 0 ↓ · flag

Owolabi Oyeyemi Rafiu · 7 days ago 🔗

Hi Philip, can you explain to me the approach for question 11 and 12.
Thanks.

↑ 0 ↓ · flag

Philip McShane · 4 days ago 🔗

Hi Owolabi, for question 11 have a look at this thread:
https://class.coursera.org/softwaresec-001/forum/thread?thread_id=233

for question 12 there is a clear explanation in this thread:
https://class.coursera.org/softwaresec-001/forum/thread?thread_id=176

↑ 0 ↓ · flag

+ Comment

Aniket Kamat · 10 days ago 🔗

so i tried and got the difference between &ptrs[0] and &p as -771675416.
So basically what I think is that to get to pat_on_back from ptrs it must advance by the values to execute pat_on_back.

am I on the right track ?

↑ 0 ↓ · flag

Riccardo Sirigu Signature Track · 10 days ago

Of course you are on the right track.
But remember to subtract the lower address from the higher one.
(By the way, &ptrs is the same as &ptrs[0])
If you can't figure out how to point to pat_on_back, try to draw the stack.. and remember that C doesn't perform bound checking..

↑ 2 ↓ · flag

Anonymous · 10 days ago

In ptrs[0] - p I got this answer as -3086701664 in normal calculator. How did you calculate this difference?

↑ 3 ↓ · flag

LEONARDO RIBEIRO · 4 days ago

I algo got this same value: -3086701664. I can't figure out how ptrs[s] is going to hit &p (0xbffff534) and call pat_on_back. If i gall ptrs[-3086701664], pat_On_back is not called.

↑ 0 ↓ · flag

Carsten Hansen · 4 days ago

In gdb when you hit the breakpoint try
p &ptrs[0]
p &ptrs[1024]
p &ptrs[1048576]
p &ptrs[268435456]
eventually you should be able to see what s should be to hit any address.

↑ 0 ↓ · flag

LEONARDO RIBEIRO · 4 days ago

Thank you guys for your help. I found out you should calculate (&p - &ptrs)/4 and enter the answer value in the running programming. Now, let's go to the others questions.

↑ 1 ↓ · flag

+ Comment

Diego E. C. Rosa · 10 days ago 🔗

Check again the comment above.

↑ 0 ↓ · flag

+ Comment

Peter · 10 days ago 🔗

something is not adding up, I understand I will need to make ptrs move up the memory till it reaches p, so logically will subtract the &ptrs from &p and this will give the address difference, if I am correct then I don't know what went wrong, I even considering dividing the answer by 4 and try it but still nothing, so ??!

↑ 0 ↓ · flag

Philip McShane · 10 days ago 🔗

In the VM theres a calculator under accessories. Open it and set it to hex do the subtraction then divide by 4 then set it to dec to see the decimal value which is what you want to give the program as input.

↑ 3 ↓ · flag

Peter · 10 days ago 🔗

this is what I did: (&p - &ptrs) / 0x4 = number
this number converted to decimal is not working

↑ 1 ↓ · flag

Imran Bashir · 10 days ago 🔗

yes same :(did you get it finally by any chance?

↑ 1 ↓ · flag

Philip McShane · 10 days ago 🔗

Try this open the calculator and make sure hex is selected.
Enter this number bffff534r (note I've removed the 0x from the start of the number)
click on dec and you should see 3221222708

If your running windows its calculator can be used for hex math by clicking view and then programmer. You should see the options for hex, dec, oct and bin on the left hand side.

↑ 1 ↓ · flag

Peter · 10 days ago 🔗

Thanks I am not sure what I was doing wrong but I am guessing in the middle of all of that I was lost moving between hex and dec and got wired values, but I got it right

↑ 0 ↓ · flag

Philip McShane · 10 days ago 🔗

Make sure you remove the 0x notation from the start of the number. This is not part of the number rather it is notation to show that the number is hex.

If you're trying to do the subtraction in gdb you'll need to use casts to unsigned int. I'm not familiar with gdb so personally I found it easier to use the calculator to do all the math.

↑ 0 ↓ · flag

+ Comment

Pedro Simões · 10 days ago 🔗

I was able to get the "Achievement unlocked!", but I'm still not 100% sure about why we need to divide by 4.

I think it's because ptrs is an array of pointers (4 byte each), so we need to jump every 4 bytes.

Am I correct or there's something else?

↑ 2 ↓ · flag

Andrew Ruef STAFF · 10 days ago 🔗

You are correct!

↑ 5 ↓ · flag

+ Comment

Eduardo Navarro Signature Track · 7 days ago 🔗

Ah simple mistake, divide by 4... 32Bit architecture uses 4 bytes for addressing... duh!

↑ 1 ↓ · flag

+ Comment

Christine M Mukai Signature Track · 7 days ago 🔗

Thank you! This discussion was very helpful!

↑ 0 ↓ · flag

+ Comment

Anonymous · 7 days ago 🔗

I have a quick question; for question 9 onwards, where and how do you input the various numbers you are trying to cause the BOF with?   Is it through the runbin script or through GDB?

TIA

↑ 0 ↓ · flag

eyan422 · 7 days ago 🔗

through the runbin script

↑ 0 ↓ · flag

+ Comment

Anonymous · 7 days ago 🔗

Reading this thread helped me find the answer without grasping why it worked. For what that's worth.

↑ 0 ↓ · flag

Owolabi Oyeyemi Rafiu · 6 days ago 🔗

Anonymous, please can you help me on Q11 and Q12 through clear explanation.
Thanks.

↑ 0 ↓ · flag

+ Comment

Anonymous · 6 days ago 🔗

Owolabi,

I'm still struggling with 9 and 10 :-(Question for others in the forum: Did we go through examples for Q 9-12 in the video lectures? I went through the videos again but couldn't find any.

🡅 0 🡇 · flag

+ Comment

Clark Wilkerson · 4 days ago 🔗

Why doesn't the answer work when I run this outside the debugger?
e.g. "./wisdom-alt" from the command line, enter the appropriate number in selection and all it does is go back to "Hello there".

If you set up the debugger as shown in Project 1 and run it from there, it will output "Achievement unlocked!" and the correct answer will be displayed but I would expect it to work both ways.

🡅 0 🡇 · flag

Seaver Johnson Milnor · 4 days ago 🔗

I have an educated guess, perhaps someone who knows better could confirm if this is correct: I think it may not work without the debugger running because the VM has to allocate memory to both processes. Running just wisdom-alt by itself would shift the memory addresses being used since GDB isn't gobbling up any memory.

🡅 0 🡇 · flag

Clark Wilkerson · 4 days ago 🔗

That's what I guessed. It makes verifying your answer a challenge.

🡅 0 🡇 · flag

Sundareswaran Selvaraj Signature Track · 4 days ago 🔗

If the program is executed without the debugger (using ./wisdom-alt) the memory address of variable "p" shifts by 8 byte. Therefore the input you need to provide should be 8 byte less of what was used while running the program using the wrapper.

🡅 0 🡇 · flag

+ Comment

Havish Mutya Signature Track · 4 days ago 🔗

HI,
i could access the address of p through ptrs .. But not sure how to execute the function. Can someone help me in this?

🡅 0 🡇 · flag

Sundareswaran Selvaraj Signature Track · 4 days ago

If you are able to access p through ptrs. I assume you are getting it through GDB use the same index as your input to the C program.

↑ 0 ↓ · flag

+ Comment

Christopher Rose · 4 days ago

Since ptrs has no bounds checking it is effectively an immense array, not a 3 element one. If the function pointer p were accessed as being in the array what would its index be? Hint: A big number.

How do you calculate the offset of a pointer/array? You need the base address of the array and the address of the function you want to access through the array. Take into account that ptrs is an array of 4 byte function pointers and calculate the offset of p as an index of ptrs.

Remember that the wisdom program takes the number you enter at the menu and uses it as an offset into ptrs. That is where you execute your target function.

↑ 1 ↓ · flag

+ Comment

ioannis bonatakis · 4 days ago

I believe I get the right number but what ever I tried so far it has failed. Let say I have aabbccdd. i give in the input \xdd\xcc\xbb\xaa ,\xaa\xbb\xcc\xdd, or the decimal as it is returned by the division but stil nothing. I have given \x8\x04\xa0\xd4+<decimal>. What should I give?

↑ 0 ↓ · flag

Philip McShane · 4 days ago

You need to input an unsigned int in decimal.

So if the answer you got was bffff534r you would enter 3221222708 as the input to the program.

↑ 1 ↓ · flag

Anonymous · 3 days ago

Hi Philip,

For Q 9, following your guidance above I got the following:
&p - &ptrs = 3221222708 - 134521044 / 4 = 3187592447 (BDFECCFF in Hex). How do I insert this into wisdom-alt? Is it as a hex or dec? I assume I type this in instead of the number 2 in the wisdom-alt menu add Wisdom. I tried both but keep getting segmentation faults ;-(

Thanks

↑ 0 ↓ · flag

Philip McShane · 3 days ago 🔗

Hi Anonymous,

You want to enter the number as dec.
You've got the right approach to solve this. Your problem is in your math, remember division has a higher precedence than subtraction.

↑ 0 ↓ · flag

Anonymous · 3 days ago 🔗

Did it!! Achievement Unlocked! Phew! And Thanks!

↑ 0 ↓ · flag

ioannis bonatakis · 3 days ago 🔗

Philip McShane Thank you. I made it

↑ 0 ↓ · flag

+ Comment

New post

To ensure a positive and productive discussion, please read our forum posting policies before posting.

B *I* Link <code> Pic Math Edit: Rich ▾ Preview

□ Make this post anonymous to other students

☑ Subscribe to this thread at the same time

Add post