Oregon State University

Computer Science

CS 444: Operating Systems II

HW: Virtualization of Memory

Author:
Jesse Piccione (piccionj@oregonstate.edu)

May 12, 2022

Goal

The goal of this project/assignment was to add the ability to cause a null pointer address exeception. In Xv6 there is no way to catch a null pointer derefence because all memory mapped in a program is mapped from start to end of the memory. Meaning when an address equals null or (0) it points to address zero at the start of the page directory. The memory is intialized and therfore does not cause an exeception. Thus change the program to make a null pointer exception upon

Procedure To Accomplish

A few things were changed in the xv6-public folder to accomplish the task:
1. change 0 to 0x1000 in Makefile

```
- $(LD) $(LDFLAGS) -N -e main -Ttext 0 -o $@ $^
+ $(LD) $(LDFLAGS) -N -e main -Ttext 0x1000 -o $@ $^
```

2. change 0 to 0x1000 in Makefile

```
-
$(LD) $(LDFLAGS) -N -e main -Ttext 0 -o _forktest forktest.o ulib.o usys.o
$(OBJDUMP) -S _forktest > forktest.asm
+
$(LD) $(LDFLAGS) -N -e main -Ttext 0x1000 -o _forktest forktest.o ulib.o usys.
  o
$(OBJDUMP) -S _forktest > forktest.asm
```

3. change sz=0 to sz=PGSIZE in exec() in exec.c

```
- sz = 0;
+ sz = PGSIZE;
  for(i=0, off=elf.phoff; i<elf.phnum; i++, off+=sizeof(ph)){
    if(readi(ip, (char*)&ph, off, sizeof(ph)) != sizeof(ph))
      goto bad;
```

4. i=0 to i=PGSIZE in loaduvm() in vm.c

```
  -  for(i = 0; i < sz; i += PGSIZE)
  +  for(i = PGSIZE; i < sz; i += PGSIZE)
```

5. change p=0 into p=4096 or PGSIZE in validatetest() in usertest.c

```
  -  for(p = 0; p <= (uint)hi; p += 4096){
  +  for(p = PGSIZE; p <= (uint)hi; p += 4096){
```

6. add in i==0 to check for null pointer address in arptr() in syscall.c

```
  -  if(size < 0 || (uint)i >= curproc->sz || (uint)i+size > curproc->sz)
  +  if(i == 0 || size < 0 || (uint)i >= curproc->sz || (uint)i+size > curproc
  ->sz)
```

Running?

To get the Updated Os to run on os2 type ”make qemu-nox” in the xv6-public file.

Reasoning Behind Solution

My solution works based on its own logic if we created leave a blank space in memory before a page the memory is not allocated. Because we leave a blank page when a programs attempts access a null pointer as an argument memory that is not aloocated for the program will be accessing memory when we should not allow. The good thing is that this will be address zero everytime. Thus we have the ability to 100 percent of the time successfully create a null pointer dereference at the proper address that is never declared (0x0).

So we are ready to catch this event when it happens. To catch the null pointer now we add in the ability into one of the system calls that checks for valid address spaces for a specific program. Simply when the address is zero it is invalid so return -1.

Testing

So to test this I complied and checked for an error using ”make qemu-nox” Then because I added a testprogam into qemu called ”FatJoey.c” I was able to run my null pointer testby typing ’FatJoey’ in qemu-nox. To get the file to work you have to add in the FatJoey.c file shown below after diffing in the patch to the xv6-public folder. Shownbelow.

```c
#include "syscall.h"
#include "types.h"
#include "user.h"

#define NULL 0

int main(){
    int a; // some integers
    int *pi;     // a pointer to an integer
    a = 5;
    pi = &a; // pi points to a
    pi = NULL; //
    printf(1,"You should not be allowed to do this: %p", *pi);
    exit();
}
```

Here is a example output:

```
init: starting sh
$ ls
.              1 1 512
..             1 1 512
```

```
README          2 2 2286
cat             2 3 22848
echo            2 4 22312
forktest        2 5 18264
grep            2 6 24108
init            2 7 22632
kill            2 8 22324
ln              2 9 22300
ls              2 10 24288
mkdir           2 11 22384
rm              2 12 22376
sh              2 13 30308
stressfs        2 14 23104
usertests       2 15 56760
wc              2 16 23168
zombie          2 17 22084
FatJoey         2 18 22608
console         3 19 0
$ FatJoey
pid 4 FatJoey: trap 14 err 4 on cpu 0 eip 0x1028 addr 0x0--kill prc
$ QEMU: Terminated
```