picoCTF Need For Speed

Points: 400.
Description: The name of the game is speed. Are you quick enough to solve this problem and keep it above 50 mph? need-for-speed.
Link: https://play.picoctf.org/practice/challenge/39?category=3&page=2

Basic Commands

```
drew@ubuntu:~/Desktop$ file need-for-speed
need-for-speed: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked,
interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0,
BuildID[sha1]=b4b1e824082c140091043151ab990149efa44806, not stripped
```

```
drew@ubuntu:~/Desktop$ strings need-for-speed
/lib64/ld-linux-x86-64.so.2
libc.so.6
exit
puts
putchar
alarm
Not fast enough. BOOM!
Something bad happened here.
Creating key...
Finished
Printing flag:
Keep this thing
GCC: (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
```

Some interesting strings, but nothing that immediately pops out at me.

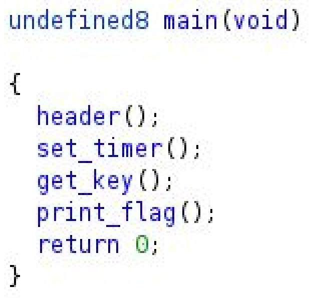
First Execution

```
drew@ubuntu:~/Desktop$ ./need-for-speed
Keep this thing over 50 mph!
============================

Creating key...
Not fast enough. BOOM!
drew@ubuntu:~/Desktop$
```

Let's look at the source code to see what is happening.

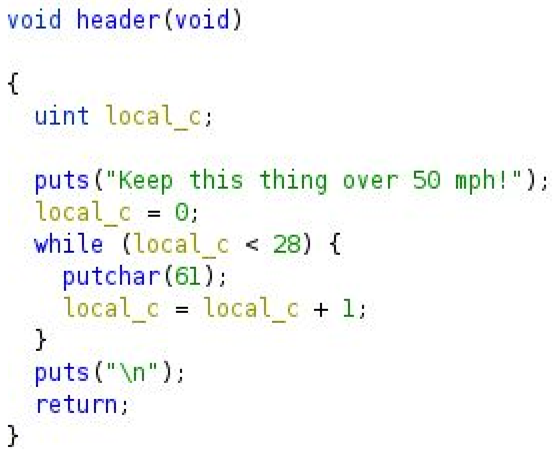
Ghidra Analysis



```
undefined8 main(void)

{
  header();
  set_timer();
  get_key();
  print_flag();
  return 0;
}
```

Multiple functions are called in main() .



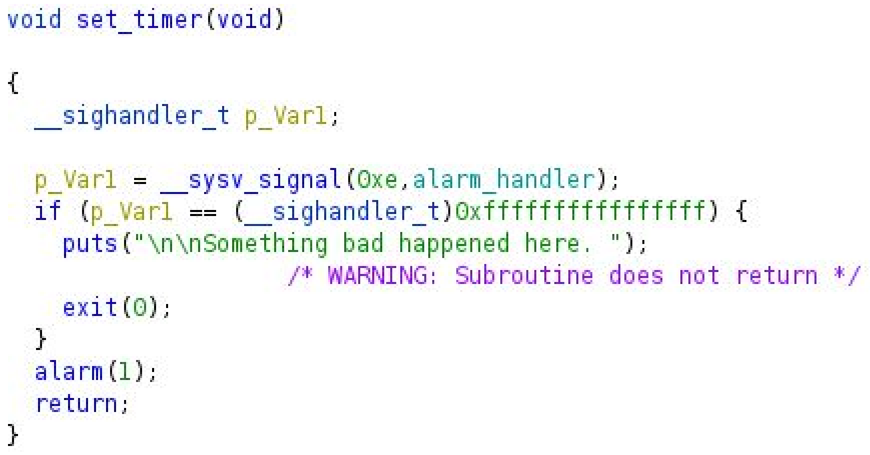
```
void header(void)

{
  uint local_c;

  puts("Keep this thing over 50 mph!");
  local_c = 0;
  while (local_c < 28) {
    putchar(61);
    local_c = local_c + 1;
  }
  puts("\n");
  return;
}
```

Local_c looks like a counter variable and while it is less than 28 it outputs an “a” to the screen.

putchar()



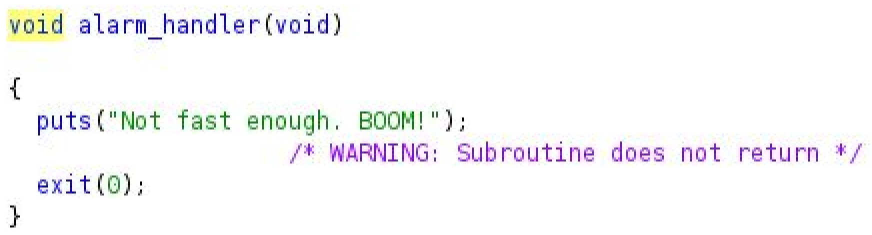
```
void set_timer(void)

{
  __sighandler_t p_Var1;

  p_Var1 = __sysv_signal(0xe,alarm_handler);
  if (p_Var1 == (__sighandler_t)0xffffffffffffffff) {
    puts("\n\nSomething bad happened here. ");
                    /* WARNING: Subroutine does not return */
    exit(0);
  }
  alarm(1);
  return;
}
```

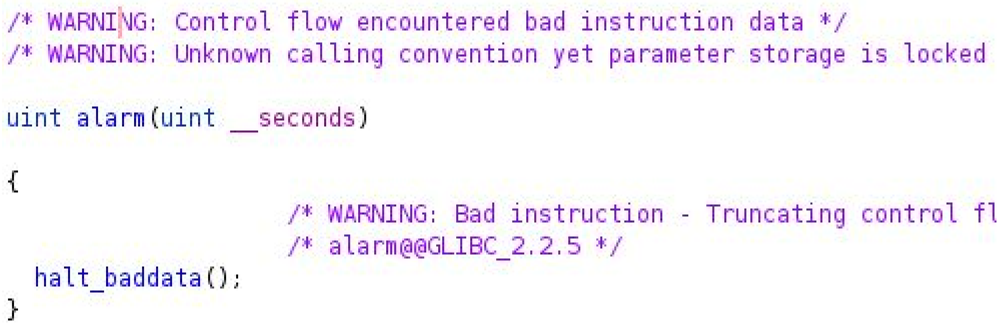
This function seems to set up a __signhandler_t variable and then proceed to call alarm().

Signals in C++



```
void alarm_handler(void)

{
  puts("Not fast enough. BOOM!");
                    /* WARNING: Subroutine does not return */
  exit(0);
}
```



```
/* WARNING: Control flow encountered bad instruction data */
/* WARNING: Unknown calling convention yet parameter storage is locked

uint alarm(uint __seconds)

{
                    /* WARNING: Bad instruction - Truncating control fl
                    /* alarm@@GLIBC_2.2.5 */
  halt_baddata();
}
```

Ghidra can not decompile the function alarm(), but I am not very concerned about it.

Understanding What's Happening

In main the function set_timer() is called which sets an alarm to end the program before the function print_flag is called. To fix this we can patch the binary so set_time() never happens.

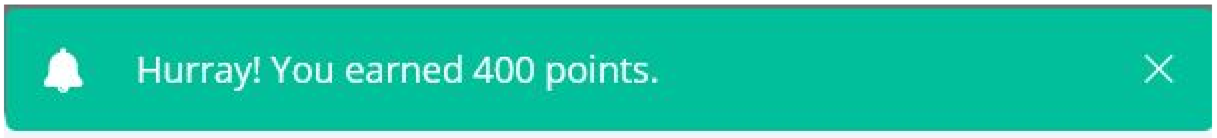
Patching The Binary Using Radare2

```
0x00000929   b800000000   mov eax, 0
0x0000092e   e8a5ffffff   call sym.header        ;[1]
0x00000933   b800000000   mov eax, 0
0x00000938   90           nop
0x00000939   90           nop
0x0000093a   90           nop
0x0000093b   90           nop
0x0000093c   90           nop
0x0000093d   b800000000   mov eax, 0
```

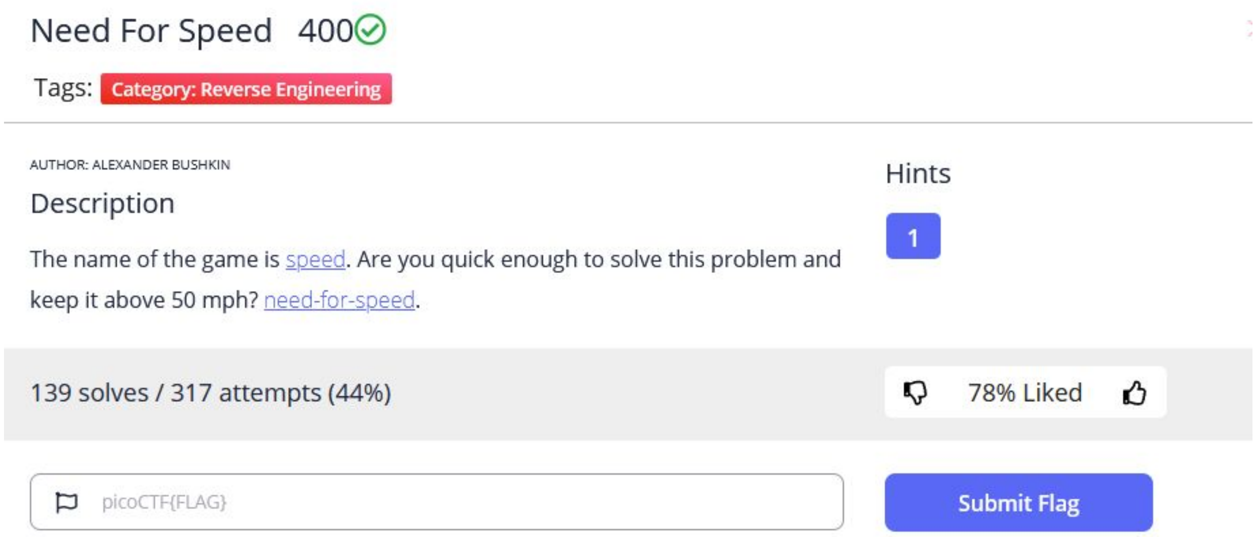
```
    0x00000942    e836ffffff     call sym.get_key       ;[2]
    0x00000947    b800000000     mov eax, 0
    0x0000094c    e85bffffff     call sym.print_flag    ;[3]
    0x00000951    b800000000     mov eax, 0
```

```
drew@ubuntu:~/Desktop$ ./need-for-speed
Keep this thing over 50 mph!
============================

Creating key...
Finished
Printing flag:
PICOCTF{Good job keeping bus #190ca38b speeding along!}
```



Hurray! You earned 400 points.



Need For Speed 400

Tags: Category: Reverse Engineering

AUTHOR: ALEXANDER BUSHKIN

Description

The name of the game is speed. Are you quick enough to solve this problem and keep it above 50 mph? need-for-speed.

Hints

1

139 solves / 317 attempts (44%)

78% Liked

picoCTF{FLAG}

Submit Flag