PWN College

Session 22

Atousa Ahsani

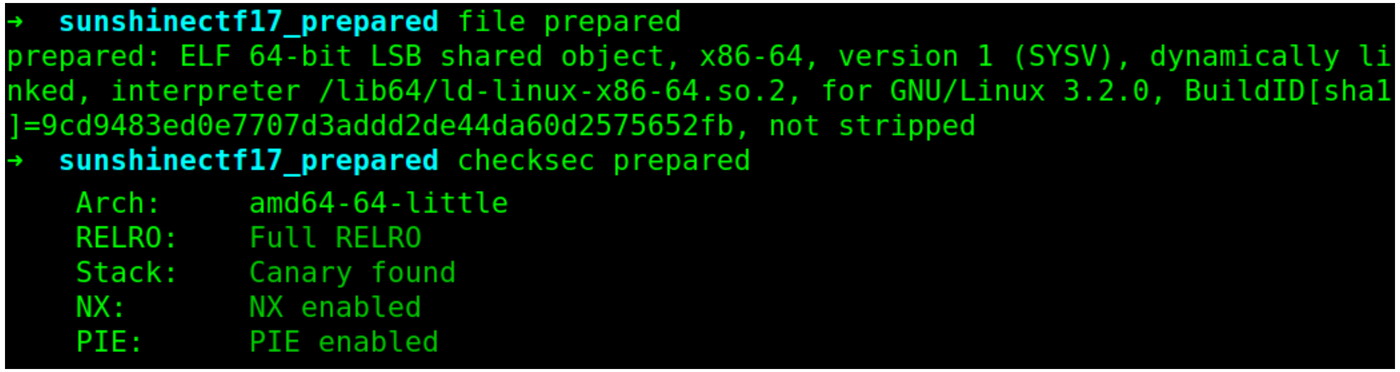
References: https://pwn.college/, https://guyinatuxedo.github.io/

Bad Seed

Sunshine CTF 2017 Prepared

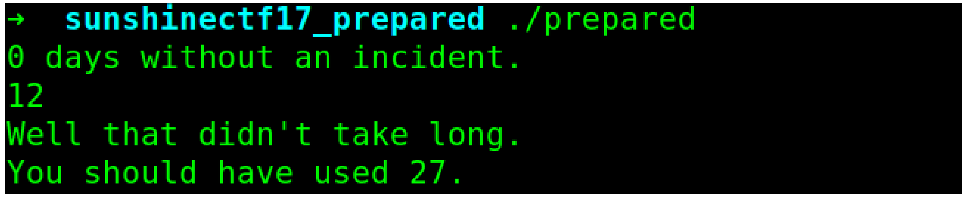
SunshineCTF'17: Prepared

- It is a **64-bit dynamically** linked binary, with a **stack canary**, non executable **stack**, and enabled **PIE**.



```
➜  sunshinectf17_prepared file prepared
prepared: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically li
nked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1
]=9cd9483ed0e7707d3addd2de44da60d2575652fb, not stripped
➜  sunshinectf17_prepared checksec prepared
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

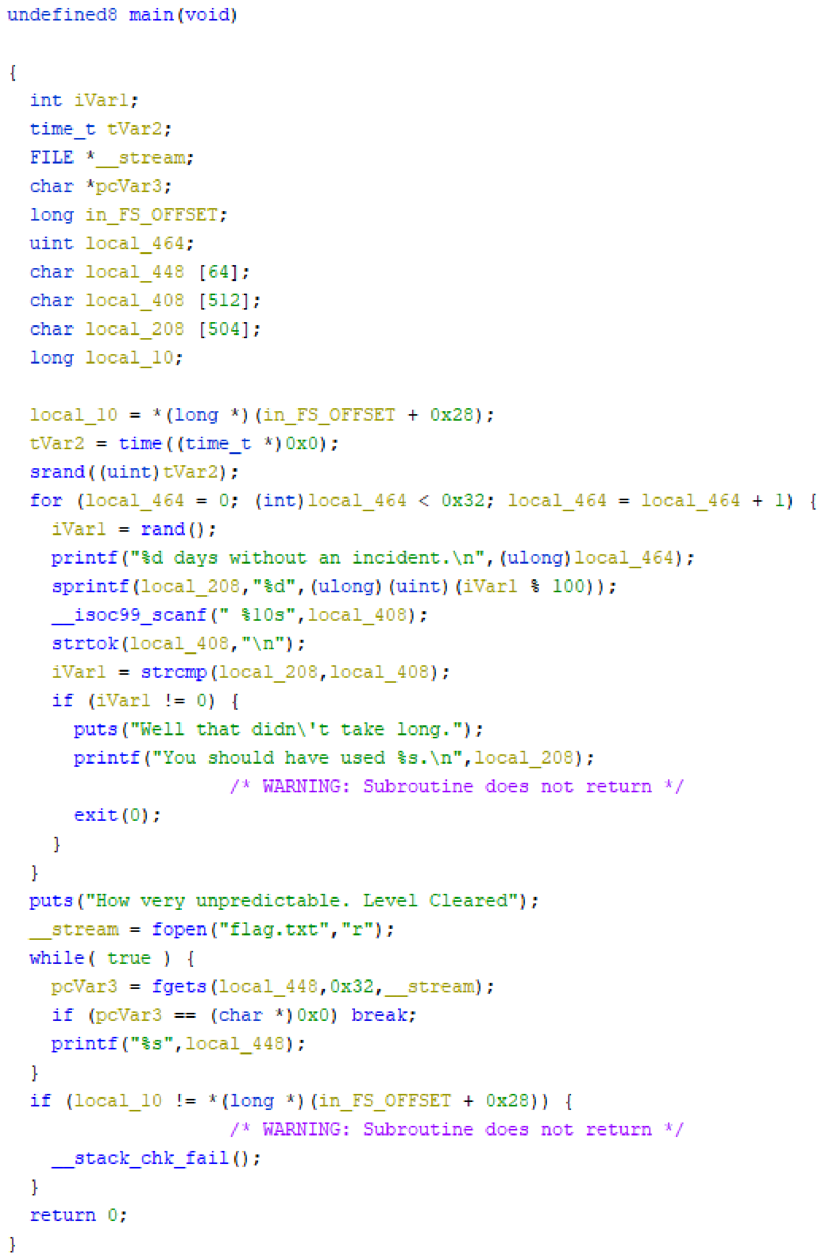
- When we run it, it prompts us for a number.



```
➜  sunshinectf17_prepared ./prepared
0 days without an incident.
12
Well that didn't take long.
You should have used 27.
```

SunshineCTF'17: Prepared

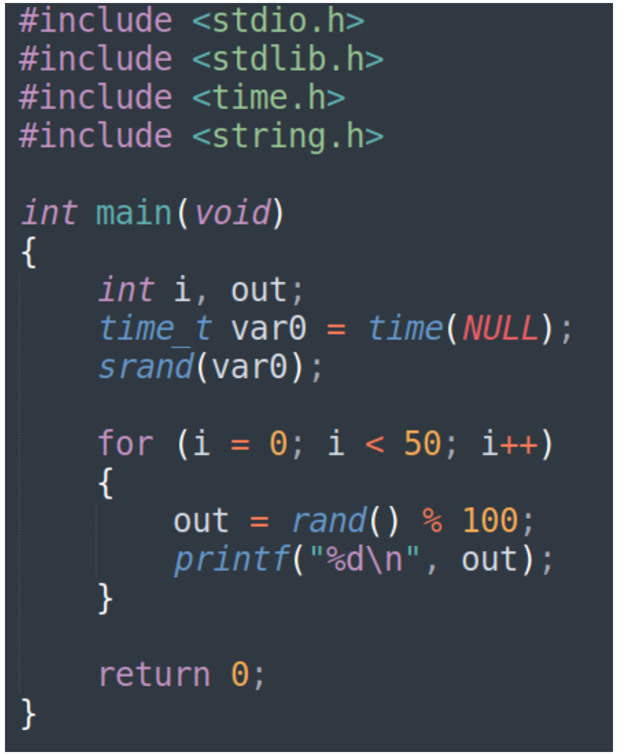
- So we can see, this is pretty similar to the other challenges in this module. It declares **time** as a ***seed*** with the ***srand*** function, then uses rand to generate values (that are modded by 100) that we have to guess in a loop that will run 50 times.
- So we have to guess what number rand will generate 50 times in a row.
- The value ***rand*** generate is directly based off of the ***seed***. So if we have the same seed, we can generate the same sequence of numbers. Also since the seed is the **current time**, we know what the seed is.



```
undefined8 main(void)

{
  int iVar1;
  time_t tVar2;
  FILE *__stream;
  char *pcVar3;
  long in_FS_OFFSET;
  uint local_464;
  char local_448 [64];
  char local_408 [512];
  char local_208 [504];
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  tVar2 = time((time_t *)0x0);
  srand((uint)tVar2);
  for (local_464 = 0; (int)local_464 < 0x32; local_464 = local_464 + 1) {
    iVar1 = rand();
    printf("%d days without an incident.\n",(ulong)local_464);
    sprintf(local_208,"%d",(ulong)(uint)(iVar1 % 100));
    __isoc99_scanf(" %10s",local_408);
    strtok(local_408,"\n");
    iVar1 = strcmp(local_208,local_408);
    if (iVar1 != 0) {
      puts("Well that didn\'t take long.");
      printf("You should have used %s.\n",local_208);
                    /* WARNING: Subroutine does not return */
      exit(0);
    }
  }
  puts("How very unpredictable. Level Cleared");
  __stream = fopen("flag.txt","r");
  while( true ) {
    pcVar3 = fgets(local_448,0x32,__stream);
    if (pcVar3 == (char *)0x0) break;
    printf("%s",local_448);
  }
  if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
  }
  return 0;
}
```

SunshineCTF'17: Prepared

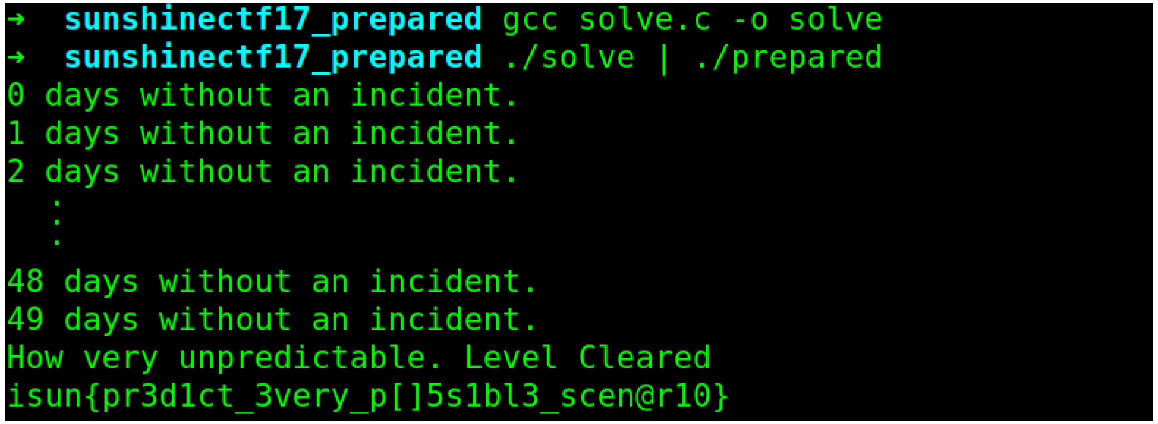
- With this we can just write a simple C program which will use **time** as a ***seed*** and generate the numbers it expects.



```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

int main(void)
{
    int i, out;
    time_t var0 = time(NULL);
    srand(var0);

    for (i = 0; i < 50; i++)
    {
        out = rand() % 100;
        printf("%d\n", out);
    }

    return 0;
}
```

SunshineCTF'17: Prepared

- We just need to compile and run the exploit code, and redirect the result to the target.



```
➜  sunshinectf17_prepared gcc solve.c -o solve
➜  sunshinectf17_prepared ./solve | ./prepared
0 days without an incident.
1 days without an incident.
2 days without an incident.
  ⋮
48 days without an incident.
49 days without an incident.
How very unpredictable. Level Cleared
isun{pr3d1ct_3very_p[]5s1bl3_scen@r10}
```