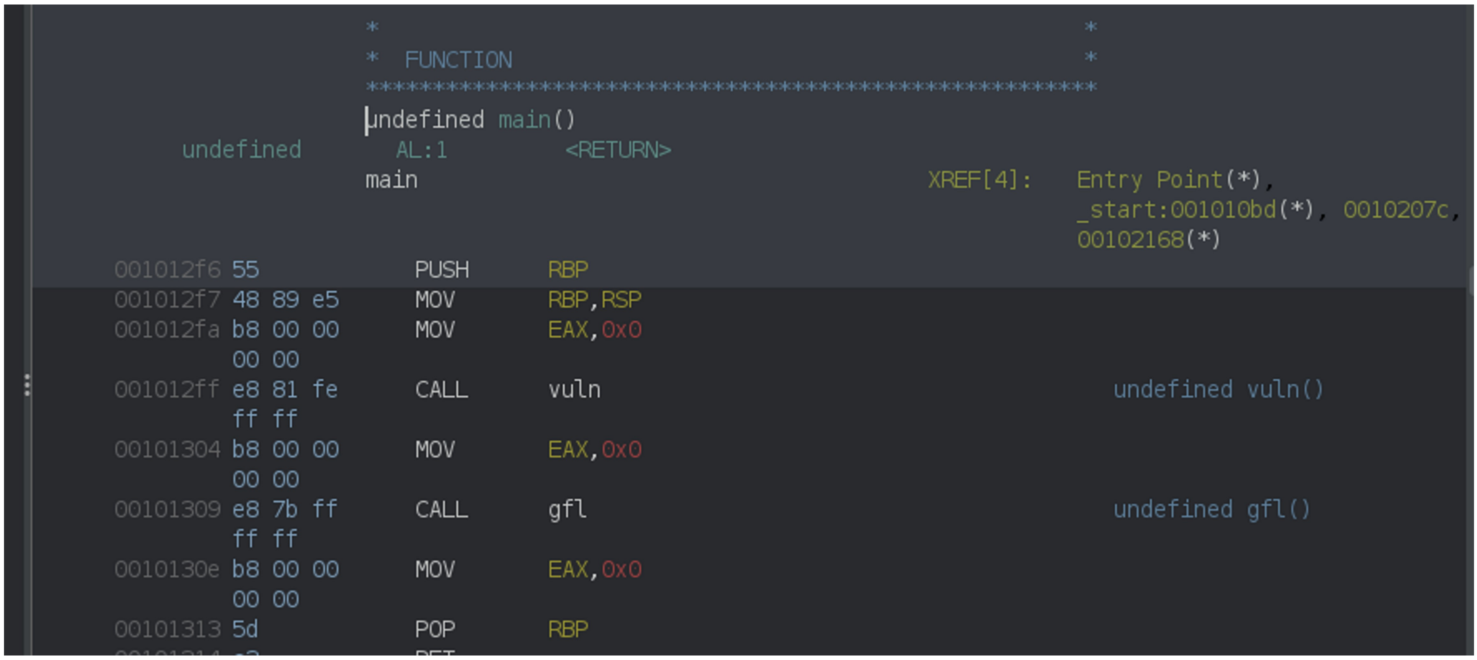
Classic Passwd

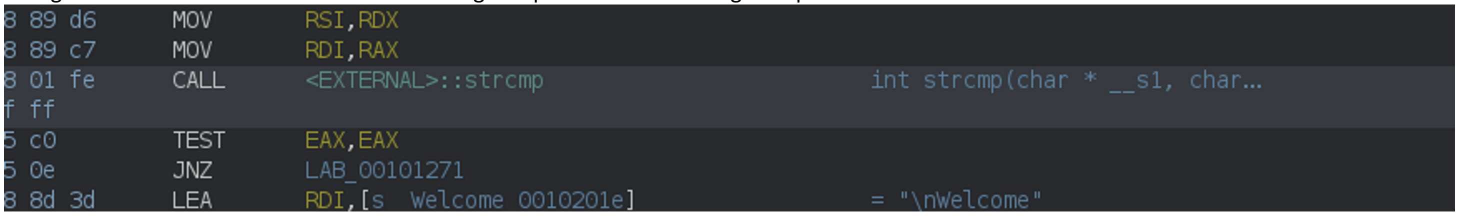
Thursday, February 17, 2022 10:31 AM

We put the program in Ghidra, and interestingly main calls two functions but doesn't seem to do anything else



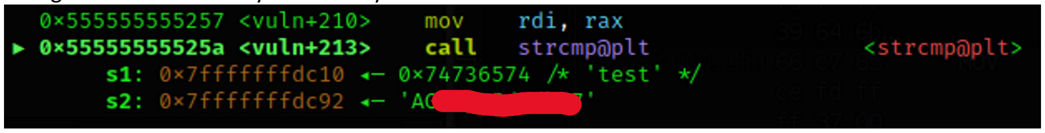
```
                            *                                                          *
                            *  FUNCTION                                                *
                            ***********************************************************
                            undefined main()
             undefined        AL:1          <RETURN>
                            main                                        XREF[4]:    Entry Point(*),
                                                                                    _start:001010bd(*), 0010207c,
                                                                                    00102168(*)
        001012f6 55              PUSH       RBP
        001012f7 48 89 e5        MOV        RBP,RSP
        001012fa b8 00 00        MOV        EAX,0x0
                 00 00
        001012ff e8 81 fe        CALL       vuln                                            undefined vuln()
                 ff ff
        00101304 b8 00 00        MOV        EAX,0x0
                 00 00
        00101309 e8 7b ff        CALL       gfl                                             undefined gfl()
                 ff ff
        0010130e b8 00 00        MOV        EAX,0x0
                 00 00
        00101313 5d              POP        RBP
```

Going to the vuln function we find an interesting comparison that should get us past the first authentication check



```
8 89 d6       MOV        RSI,RDX
8 89 c7       MOV        RDI,RAX
8 01 fe       CALL       <EXTERNAL>::strcmp                         int strcmp(char * __s1, char...
f ff
5 c0          TEST       EAX,EAX
5 0e          JNZ        LAB_00101271
8 8d 3d       LEA        RDI,[s_Welcome_0010201e]                   = "\nWelcome"
```

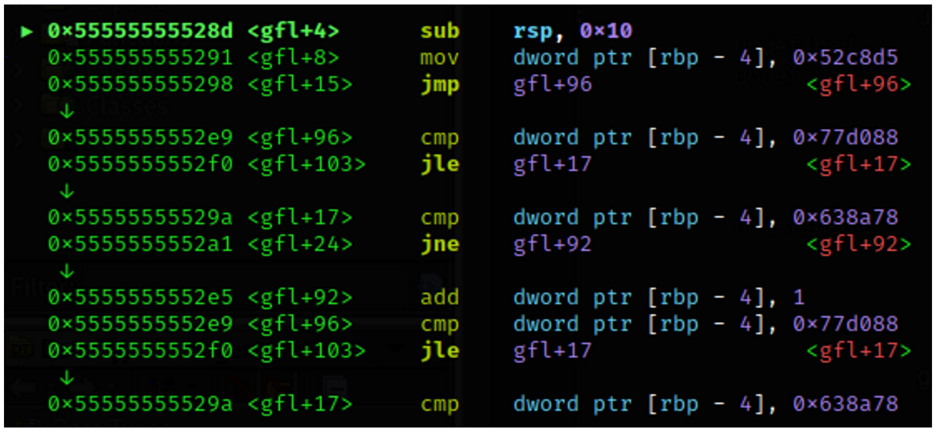
Going to that call in the dynamic analysis leaks the username



```
   0x555555555257 <vuln+210>    mov    rdi, rax
 ► 0x55555555525a <vuln+213>    call   strcmp@plt                         <strcmp@plt>
        s1: 0x7fffffffdc10 ◂— 0x74736574 /* 'test' */
        s2: 0x7fffffffdc92 ◂— 'AC[redacted]'
```

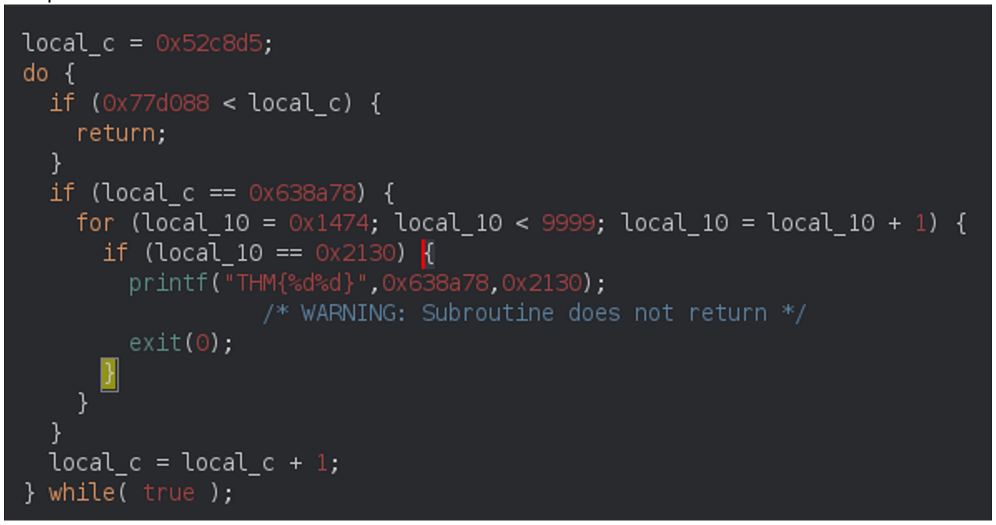
AG[redacted]

That gets us to the gfl() function



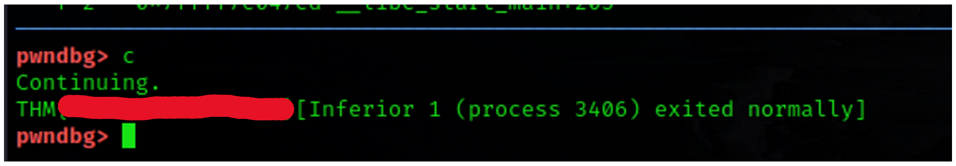
```
► 0x55555555528d <gfl+4>      sub    rsp, 0x10
  0x555555555291 <gfl+8>      mov    dword ptr [rbp - 4], 0x52c8d5
  0x555555555298 <gfl+15>     jmp    gfl+96                         <gfl+96>
   ↓
  0x5555555552e9 <gfl+96>     cmp    dword ptr [rbp - 4], 0x77d088
  0x5555555552f0 <gfl+103>    jle    gfl+17                         <gfl+17>
   ↓
  0x55555555529a <gfl+17>     cmp    dword ptr [rbp - 4], 0x638a78
  0x5555555552a1 <gfl+24>     jne    gfl+92                         <gfl+92>
   ↓
  0x5555555552e5 <gfl+92>     add    dword ptr [rbp - 4], 1
  0x5555555552e9 <gfl+96>     cmp    dword ptr [rbp - 4], 0x77d088
  0x5555555552f0 <gfl+103>    jle    gfl+17                         <gfl+17>
   ↓
  0x55555555529a <gfl+17>     cmp    dword ptr [rbp - 4], 0x638a78
```

Based on the decompilation in Ghidra, it looks like we can get passed this and print the flag if we just bypass all the comparisons



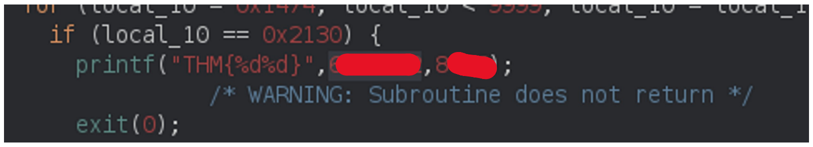
```
local_c = 0x52c8d5;
do {
  if (0x77d088 < local_c) {
    return;
  }
  if (local_c == 0x638a78) {
    for (local_10 = 0x1474; local_10 < 9999; local_10 = local_10 + 1) {
      if (local_10 == 0x2130) {
        printf("THM{%d%d}",0x638a78,0x2130);
                    /* WARNING: Subroutine does not return */
        exit(0);
      }
    }
  }
  local_c = local_c + 1;
} while( true );
```

So we should be able to set the registers to satisfy the comparisons and walk our way through this function



```
pwndbg> c
Continuing.
THM[redacted][Inferior 1 (process 3406) exited normally]
pwndbg> 
```

Interestingly, converting the decompiled function values to ints proved to be the correct flag as well



```
for (local_10 = 0x1474; local_10 < 9999; local_10 = local_1
  if (local_10 == 0x2130) {
    printf("THM{%d%d}",[redacted],[redacted]);
                /* WARNING: Subroutine does not return */
    exit(0);
```