Buffer Overflow, 4ème partie

1. Point de départ : code source d'un serveur vulnérable, vsrv.c

```
/* vsrv.c */
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

#define MAXBUF  768
#define M       512

char s[MAXBUF];

void mystrcpy(char *dest, char *src) {
  int i, l = strlen(src);
  for (i=0; src[i] != '\0' && i<MAXBUF; i++)
    dest[i] = src[i];
  dest[i]='\0';
}

void myoutput(char *str) {
  char msg[M];

  printf("&msg=0x%08x\n",&msg);
  mystrcpy(msg, str);
  printf("%s\n", msg);
}

int main(void) {
  setvbuf(stdout, NULL, _IONBF, 0);
  myoutput("hello");
  while (fgets(s, MAXBUF, stdin) != NULL) {
    myoutput(s);
  }
}
```

Le programme se compile avec le fichier Makefile suivant :

```
EXE = vsrv
CC=gcc
CFLAGS=-m32 -O0 -fno-stack-protector -z execstack \
-mpreferred-stack-boundary=2 -D_FORTIFY_SOURCE=0 \
-static -g

all: $(EXE)
```

2. Tout d'abord, où se trouve la vulnérabilité de type buffer overflow ? Exploiter cette vulnérabilité du programme localement.
3. Objectif : Une fois que l'exploitation de la vulnérabilité fonctionne localement, exploiter la même vulnérabilité à distance.

   Le dispositif : un service identique est à l'écoute sur une adresse IP et un port donné

   NB : le port est différent pour chacun groupe ; le port sera indiqué au tableau.

   Il s'agit d'exploiter la vulnérabilité en se connectant sur l'adresse et le port indiqués et en envoyant une chaîne susceptible de provoquer un buffer overflow. Quel procédé utiliser pour interagir avec le shell une fois que le débordement a réussi ?

   Une fois qu'un shell a été exécuté, un fichier nommé "secretN.txt" est accessible (N correspond au numéro du groupe). Quel mot de passe contient-il ?