INSA CVL
STI
2SU-5A
Sokhna Mai FALL

M.KAUFFMANN

TD2 : EMILY

- Ajout des librairies

include <stdio.h>
#include <stdlib.h>
#include <string.h>

- Programme

```
  GNU nano 3.2

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int is_valid(const char* password)
{
    if (strcmp(password, "poop") == 0) {
        return 1;
    } else {
        return 0;
    }
}

int main()
{
    char* input = NULL;
    input = malloc(256);
    printf("Please input a word: ");
    scanf("%s", input);

    if (is_valid(input)) {
        printf("That's correct!\n");
    } else {
        printf("That's not correct!\n");
    }

    free(input);
    return 0;
}
```

- Compilation et execution

```
user@optiplex-1504:~/Sec_Emb$ nano program.c
user@optiplex-1504:~/Sec_Emb$ gcc program.c -o program
user@optiplex-1504:~/Sec_Emb$ ./program
Please input a word: poop
That's correct!
user@optiplex-1504:~/Sec_Emb$ ./program
Please input a word: popp
That's not correct!
```

- Description binaire du fichier

```
user@optiplex-1504:~/Sec_Emb$ file program
program: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=0808406ec0841a7522651640b7c2e8a885057baa, not st
ipped
```

- Affichage du fichier en hexa

```
user@optiplex-1504:~/Sec_Emb$ hexdump -C program | head -n 20
00000000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  |.ELF............|
00000010  03 00 3e 00 01 00 00 00  a0 10 00 00 00 00 00 00  |..>.............|
00000020  40 00 00 00 00 00 00 00  88 3a 00 00 00 00 00 00  |@........:......|
00000030  00 00 00 00 40 00 38 00  0b 00 40 00 1e 00 1d 00  |....@.8...@.....|
00000040  06 00 00 00 04 00 00 00  40 00 00 00 00 00 00 00  |........@.......|
00000050  40 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  |@.......@.......|
00000060  68 02 00 00 00 00 00 00  68 02 00 00 00 00 00 00  |h.......h.......|
00000070  08 00 00 00 00 00 00 00  03 00 00 00 04 00 00 00  |................|
00000080  a8 02 00 00 00 00 00 00  a8 02 00 00 00 00 00 00  |................|
00000090  a8 02 00 00 00 00 00 00  1c 00 00 00 00 00 00 00  |................|
000000a0  1c 00 00 00 00 00 00 00  01 00 00 00 00 00 00 00  |................|
000000b0  01 00 00 00 04 00 00 00  00 00 00 00 00 00 00 00  |................|
000000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000d0  a0 06 00 00 00 00 00 00  a0 06 00 00 00 00 00 00  |................|
000000e0  00 10 00 00 00 00 00 00  01 00 00 00 05 00 00 00  |................|
000000f0  00 10 00 00 00 00 00 00  00 10 00 00 00 00 00 00  |................|
00000100  00 10 00 00 00 00 00 00  ad 02 00 00 00 00 00 00  |................|
00000110  ad 02 00 00 00 00 00 00  00 10 00 00 00 00 00 00  |................|
00000120  01 00 00 00 04 00 00 00  00 20 00 00 00 00 00 00  |......... ......|
00000130  00 20 00 00 00 00 00 00  00 20 00 00 00 00 00 00  |. ....... ......|
```

```
00001150  00 00 48 89 e5 74 0c 48  8b 3d f2 2e 00 00 e8 2d  |..H..t.H.=.....-|
00001160  ff ff ff e8 68 ff ff ff  c6 05 e9 2e 00 00 01 5d  |....h..........]|
00001170  c3 0f 1f 80 00 00 00 00  c3 0f 1f 80 00 00 00 00  |................|
00001180  e9 7b ff ff ff 55 48 89  e5 48 83 ec 10 48 89 7d  |.{...UH..H...H.}|
00001190  f8 48 8b 45 f8 48 8d 35  68 0e 00 00 48 89 c7 e8  |.H.E.H.5h...H...|
000011a0  bc fe ff ff 85 c0 75 07  b8 01 00 00 01 eb 01 b8  |......u.........|
000011b0  00 00 00 00 c9 c3 55 48  89 e5 48 83 ec 10 48 c7  |......UH..H...H.|
000011c0  45 f8 00 00 00 00 bf 00  01 00 00 e8 a0 fe ff ff  |E...............|
000011d0  48 89 45 f8 48 8d 3d 2e  0e 00 00 b8 00 00 00 00  |H.E.H.=.........|
000011e0  e8 6b fe ff ff 48 8b 45  f8 48 89 c6 48 8d 3d 2c  |.k...H.E.H..H.=,|
000011f0  0e 00 00 b8 00 00 00 00  e8 83 fe ff ff 48 8b 45  |.............H.E|
00001200  f8 48 89 c7 e8 7c ff ff  ff 85 c0 74 0e 48 8d 3d  |.H...|.....t.H.=|
00001210  0e 0e 00 00 e8 27 fe ff  ff eb 0c 48 8d 3d 10 0e  |.....'.....H.=..|
00001220  00 00 e8 19 fe ff ff 48  8b 45 f8 48 89 c7 e8 fd  |.......H.E.H....|
00001230  fd ff ff b8 00 00 00 00  c9 c3 66 0f 1f 44 00 00  |..........f..D..|
```

```
0000000000001185 <is_valid>:
    1185:       55                      push   %rbp
    1186:       48 89 e5                mov    %rsp,%rbp
    1189:       48 83 ec 10             sub    $0x10,%rsp
    118d:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
    1191:       48 8b 45 f8             mov    -0x8(%rbp),%rax
    1195:       48 8d 35 68 0e 00 00    lea    0xe68(%rip),%rsi        # 2004 <_IO_stdin_used+0x4>
    119c:       48 89 c7                mov    %rax,%rdi
    119f:       e8 bc fe ff ff          callq  1060 <strcmp@plt>
    11a4:       85 c0                   test   %eax,%eax
    11a6:       75 07                   jne    11af <is_valid+0x2a>
    11a8:       b8 01 00 00 01          mov    $0x1000001,%eax
    11ad:       eb 01                   jmp    11b0 <is_valid+0x2b>
    11af:       b8 00 00 00 00          mov    $0x1000000,%eax
    11b4:       c9                      leaveq
    11b5:       c3                      retq
```

```
11ad:       eb 01                   jmp
11af:       b8 00 00 00 00          mov
11b4:       c9                      leaveq
11b5:       c3                      retq
```

Nous permettant d'avoir l'emplacement binaire du fichier

- Modification du fichier

```
user@optiplex-1504:~/Sec_Emb$ printf '\x01' | dd of=program bs=1 seek=4531 count=1 conv=notrunc
1+0 enregistrements lus
1+0 enregistrements écrits
1 octet copié, 0,000106225 s, 9,4 kB/s
user@optiplex-1504:~/Sec_Emb$
```

- Visualisation des changements

```
user@optiplex-1504:~/Sec_Emb$ ./program
Please input a word: mmmm
That's correct!
user@optiplex-1504:~/Sec_Emb$ ./program
Please input a word: butts
That's correct!
user@optiplex-1504:~/Sec_Emb$
```

Conclusion

Les concepts ici s'appliquent également à comprendre combien d'exploits de sécurité fonctionnent à un niveau mécanique comme le craquage de logiciels.

Questions :

Préambule :

- Utilisation de Ghidra

  Ghidra est un logiciel libre d'ingénierie inverse développé par la NSA. Son interface graphique intègre un désassembleur et un décompilateur afin de réaliser l'analyse de fichiers binaires.

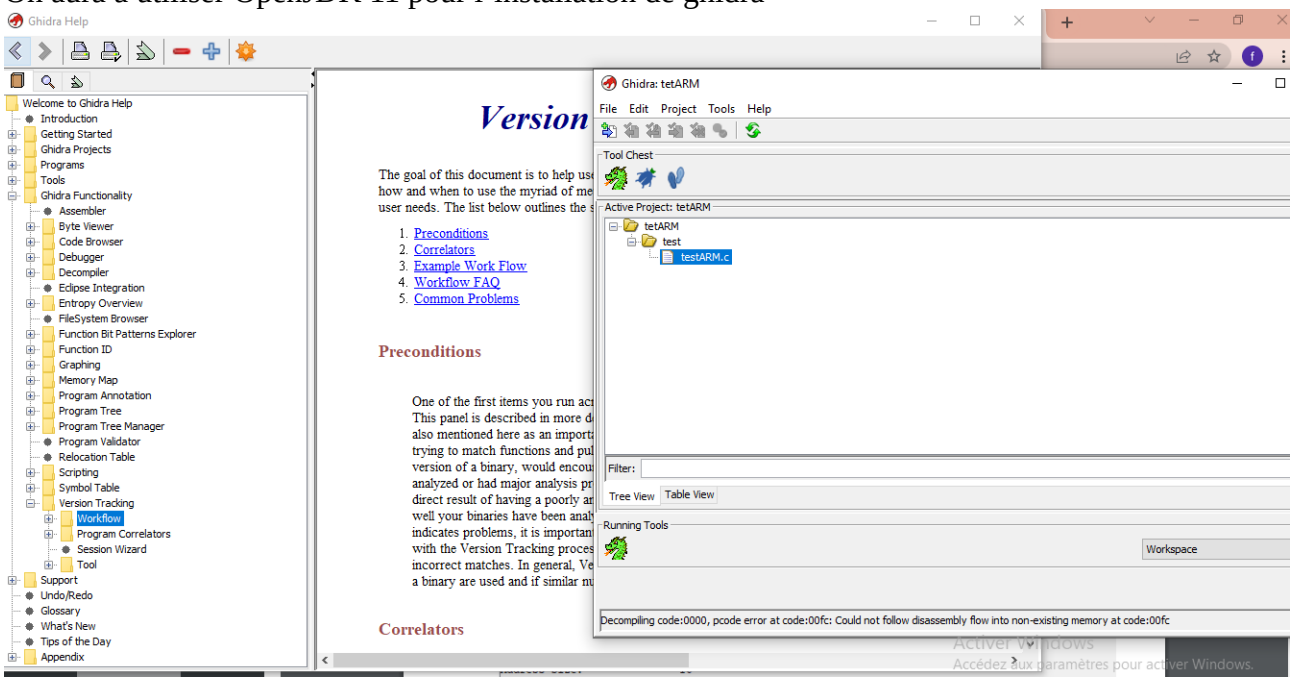
Avantages :

L'un des avantages est que l'installation de Ghidra à des fins personnelles ne nécessite pas d'avoir les droits administrateurs. De plus, comme l'installation de Ghidra ne met à jour aucune

configuration de système d'exploitation telle que le registre sous Windows, la suppression de Ghidra est aussi simple que la suppression du répertoire d'installation de Ghidra.
Outre un guide d'installation, la documentation de Ghidra comprend également des cours et des exercices pour les débutants, les niveaux intermédiaires et les niveaux avancés qui aideront les utilisateurs à s'habituer à l'interface graphique de l'outil, qui est très différente de tout outil similaire.

- Installation Ghidra

On aura à utiliser OpenJDK 11 pour l'installation de ghidra



1. Patcher le binaire. Quelle différence dans un environnement ARM ?

Dans l'environnement ARM, le paquetage binutils
contient des outils utilisés pour la génération et la
manipulation des exécutables ou des fichiers objets (.o) intermédiaires
ces différents outils, ainsi que le compilateur, seront exécutés
dans un environnement x86 (Linux ou Windows) mais le code généré sera d'un type différent.

Travaux Pratique :

On aura à travailler sur VsCode et Ghidra .

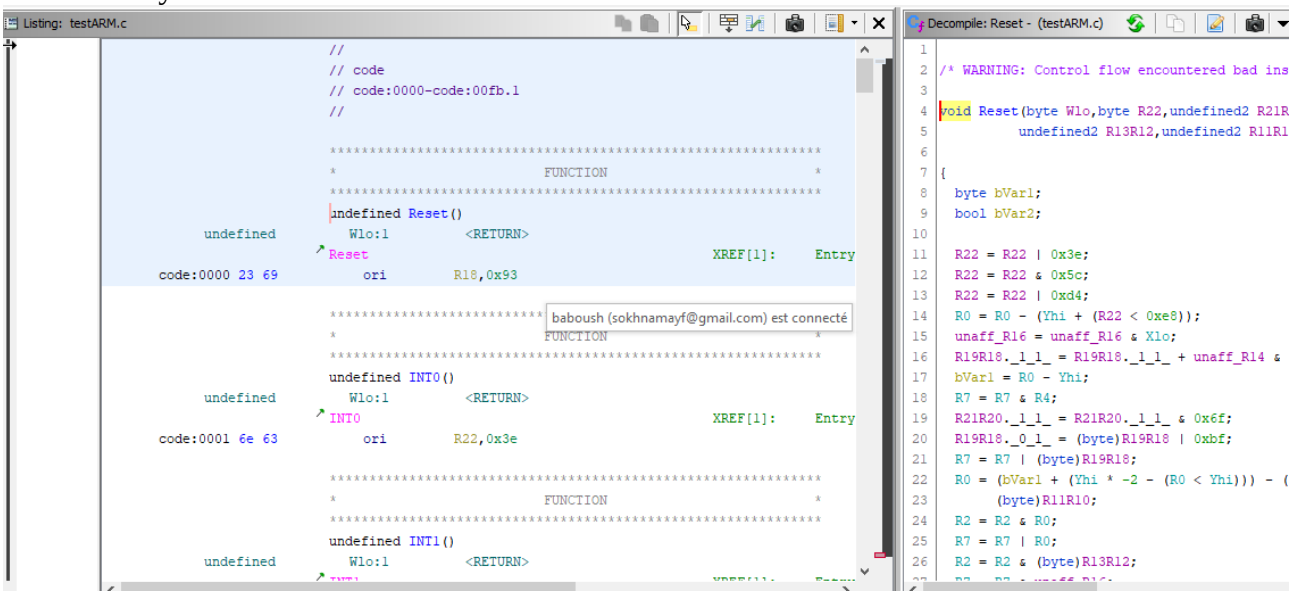
- Programme en C

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

int is_valid(const char* password)
{
    if (strcmp(password, "poop") == 0) {
        return 1;
    } else {
        return 0;
    }
}
int main()
{
    char* input = NULL;
    input = malloc(256);3
    printf("Please input a word: ");
    scanf("%s", input);

    if (is_valid(input)) {
        printf("That's correct!\n");
    } else {
        printf("That's not correct!\n");
    }

    free(input);
    return 0;
}
```

- Description de notre fichier sur ghidra

```
Project File Name:          testARM.c
Last Modified:              Thu Feb 03 01:08:10 CET 2022
Readonly:                   false
Program Name:               testARM.c
Language ID:                avr8:LE:16:default (1.2)
Compiler ID:                gcc
Processor:                  AVR8
Endian:                     Little
Address Size:               16
Minimum Address:            code:0000
Maximum Address:            mem:0fff
# of Bytes:                 4584
# of Memory Blocks:         4
# of Instructions:          0
# of Defined Data:          0
# of Functions:             0
# of Symbols:               177
# of Data Types:            0
# of Data Type Categories:  1
Created With Ghidra Version:10.0.1
Date Created:               Thu Feb 03 01:08:10 CET 2022
Executable Format:          Raw Binary
Executable Location:        /C:/Users/ibrahim/Documents/testARM.c
Executable MD5:             8c62d22ca8df3528081fa9217a9350c4
Executable SHA256:          5e97560477fdffe340d990dc6aaef087d2519256eaf09eee071179ce493d7fca
FSRL:                       file:///C:/Users/ibrahim/Documents/testARM.c?MD5=8c62d22ca8df3528081f
```

- Analyse de notre fichier



2. Comprendre le lien les attaques physiques / expliquez quelles sont les attaques par patching possibles sur une boucle for.

La sécurité physique vise à favoriser l'exploitation des équipements informatiques dans des conditions fonctionnelles optimales, de manière à bénéficier d'un maximum de performances durant un maximum de temps, Donc l'attaque physique se fait au niveau de l'équipement direct.
On note les attaques suivants : sur le local informatique (Data Center), mesure organisationnel de sécurité et les incidents (feu,eaux…).

3. Qu'elle défense est ce que je peux utiliser contre le patching ?

La définition des filtres d'entrée supprime les exploits du flux d'entrée, permettant ainsi à l'application vulnérable de continuer à fonctionner normalement même en cas d'attaque. Les filtres d'entrée générés sont garantis pour filtrer uniquement les exploits, donc sûrs à déployer automatiquement.