8INF857 – Sécurité informatique – Devoir sur la pile
Lecoq Simon

La stack frame lorsque l'exécution du code atteint le breakpoint est la suivante :

Stack frame	Adresse	Description abstraite	Valeur concrète
main()	0xBFFFFFFC	Adresse de retour de main	Non précisé (vers du code OS)
	0xBFFFFFF8	EBP précédent	Non précisé
	0xBFFFFFF4	int x *(variable locale)*	5
	0xBFFFFFF0	char *s *(variable locale)*	0x12345678
bar()	0xBFFFFFEC	int c *(argument)*	4
	0xBFFFFFE8	char *s *(argument)*	0xBFFFFFF0
	0xBFFFFFE4	int a *(argument)*	5
	0xBFFFFFE0	Adresse de la ligne 7	Non précisé
	0xBFFFFFDC	EBP précédent	0xBFFFFFF8
	0xBFFFFFD8	int d *(variable locale)*	3
	0xBFFFFFD4		
	0xBFFFFFD0		
	0xBFFFFFC8		
	0xBFFFFFC4		
	0xBFFFFFC0		

La stack frame de l'appel à foo() (représentée ci-dessous) est dépilée lorsque la ligne 6 commence à être traitée.

Stack frame	Adresse	Description abstraite	Valeur concrète
foo()	0xBFFFFFEC	int a *(argument)*	5
	0xBFFFFFE8	Adresse de la ligne 6	Non précisé
	0xBFFFFFE4	EBP précédent	0xBFFFFFF8

Le code exécuté est le suivant :

```
1  void main()
2  {
3      int x = 5;
4      char *s = (char *)malloc(64 * sizeof(char));
5      foo(x);
6      bar(x, s, 4);
7  }
8  int foo(int a)
9  {
10     return a * 4;
11 }
12 int bar(int a, char* s, int c)
13 {
14     int d = 3;
15     if (s != NULL)
16         return a + c + d;
17 }
```