Buffer overflow

Tao "DePierre" S.

Le hackerspace de Belfort

8 mai 2013

Plan

1. Définition
2. En mémoire
   - Vue générale
   - Etude de la stack
   - Récapitulatif
3. Smash the stack
   - Stratégie
   - Méthode
4. Exemple
   - Exploitation d'un buffer overflow
5. Questions

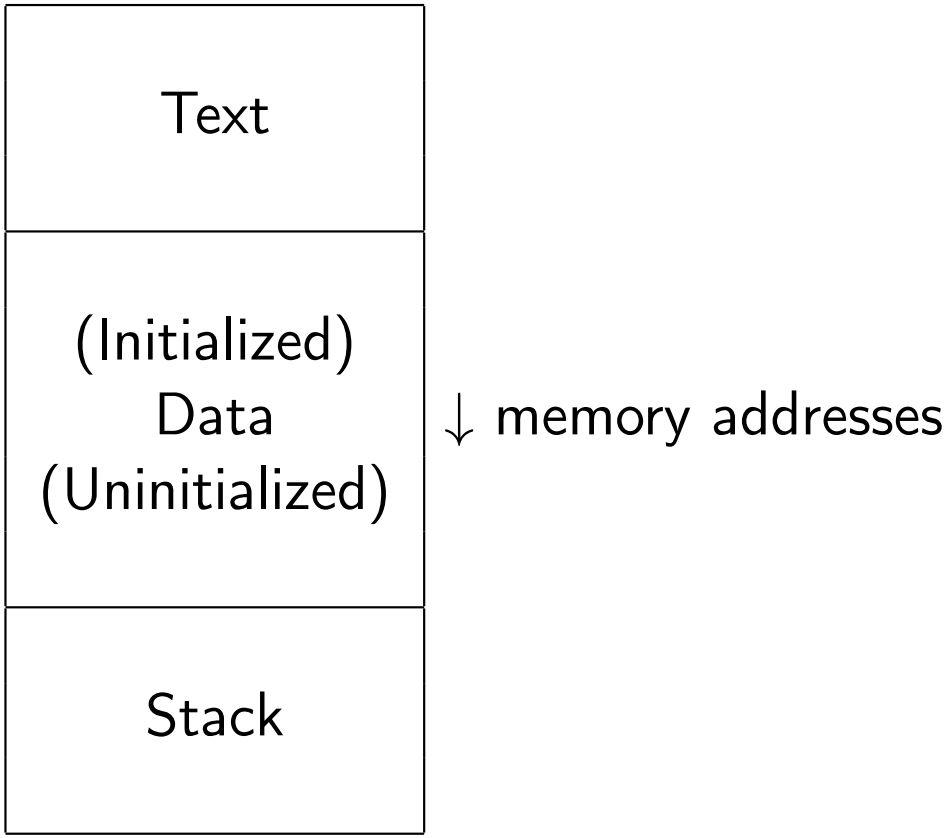
Définition formelle

"In computer security and programming, a buffer overflow, or buffer overrun, is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory."
(Wikipedia)

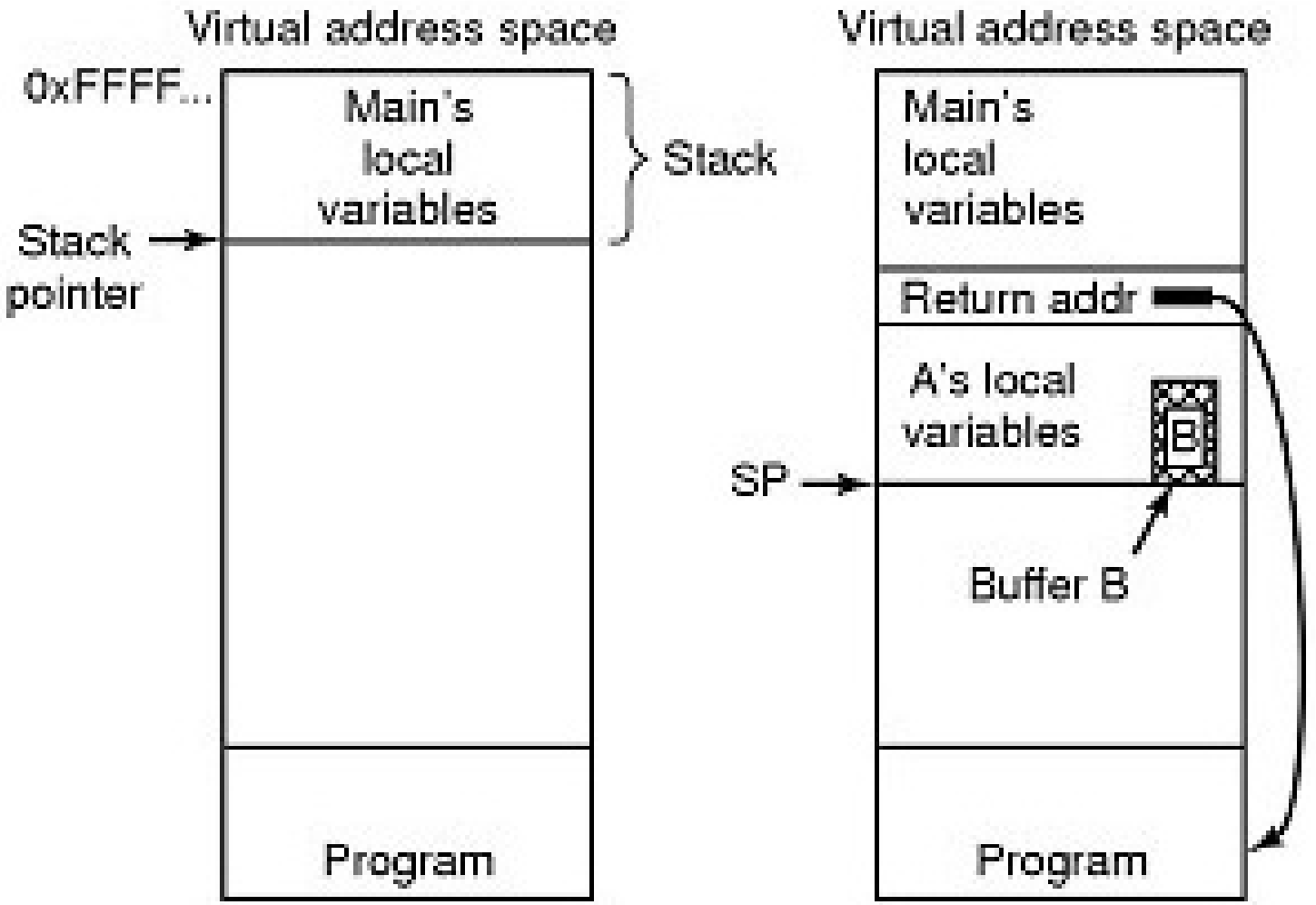
Buffer overflow

1. Définition
2. En mémoire
   - Vue générale
   - Etude de la stack
   - Récapitulatif
3. Smash the stack
   - Stratégie
   - Méthode
4. Exemple
   - Exploitation d'un buffer overflow
5. Questions

Vue générale



Fonctionnement de la stack



Buffer overflow

1. Définition
2. En mémoire
   - Vue générale
   - Etude de la stack
   - Récapitulatif
3. Smash the stack
   - Stratégie
   - Méthode
4. Exemple
   - Exploitation d'un buffer overflow
5. Questions

Exemple 1 (1/3)

```
1 void function(int a, int b, int c) {
2     char buffer[20];
3 }
4
5 int main(int argv, char *argc[]) {
6     function(1, 2, 3);
7     return 0;
8 }
```

Exemple 1 (2/3)

La fonction 'main'

```
1  push   ebp
2  mov    ebp,esp
3  sub    esp,0xc
4  mov    DWORD PTR [esp+0x8],0x3
5  mov    DWORD PTR [esp+0x4],0x2
6  mov    DWORD PTR [esp],0x1
7  call   0x80483d0 <function>
8  mov    eax,0x0
9  leave
10 ret
```

Exemple 1 (3/3)

La fonction 'function'

```
1 push    ebp ; Push the frame pointer onto the stack
2 mov     ebp,esp ; EBP now contains the new frame pointer
3 sub     esp,0x20 ; Allocate space for the local variables
4 leave
5 ret
```

Exemple 2

La nouvelle fonction 'function'

```
1 void function() {
2     int i = 0;
3     int tabint[20] = {0};
4     for (i = 0; i < 20; i = i + 1)
5         tabint[i] = i;
6 }
```

L'état de la stack après son exécution

```
1 0xffffd8d8: 0x00000001 0x00000002 0x00000003 0x00000004
2 0xffffd8e8: 0x00000005 0x00000006 0x00000007 0x00000008
3 0xffffd8f8: 0x00000009 0x0000000a 0x0000000b 0x0000000c
4 0xffffd908: 0x0000000d 0x0000000e 0x0000000f 0x00000010
5 0xffffd918: 0x00000011 0x00000012 0x00000013 0x00000014
```

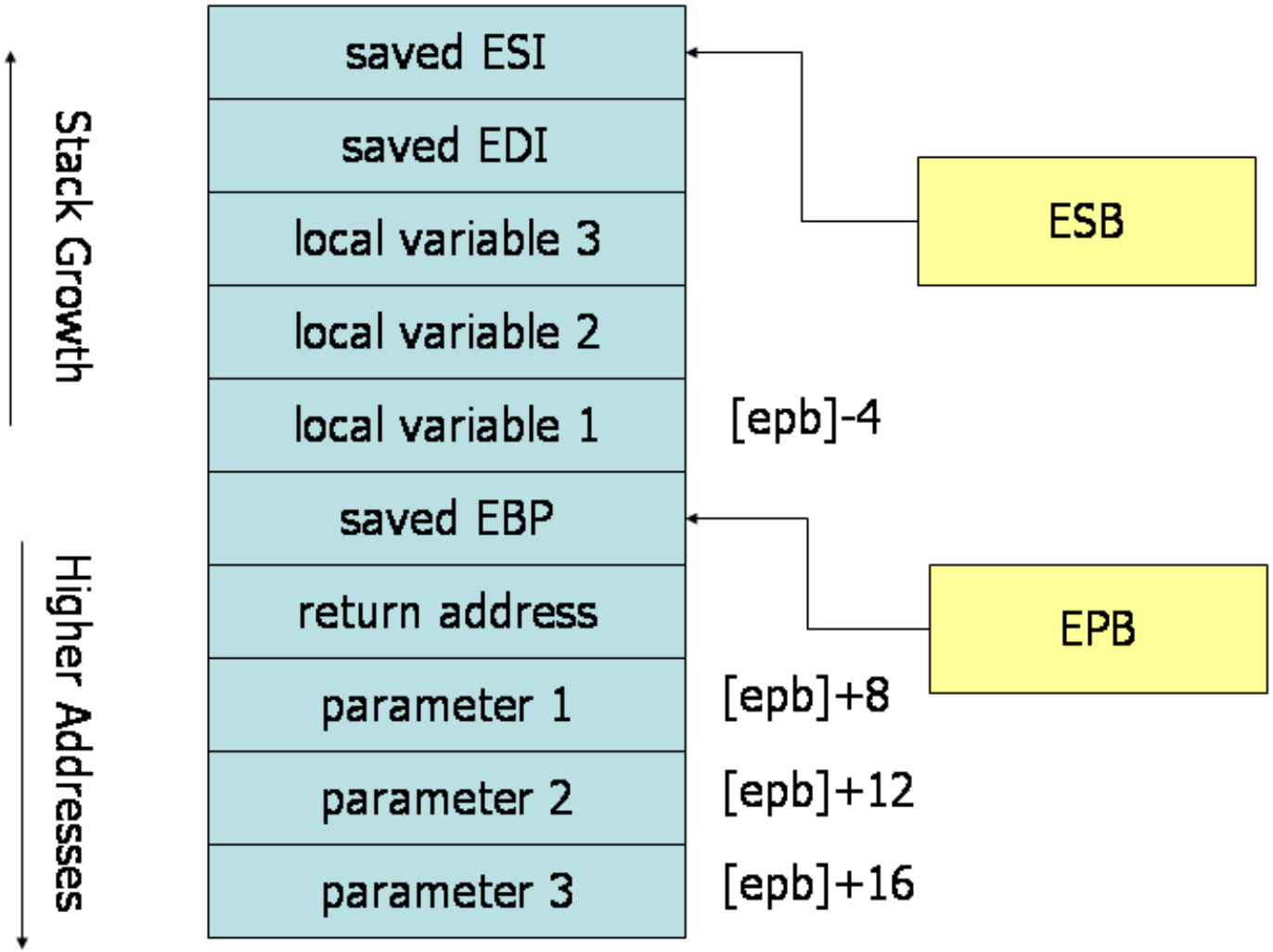
Buffer overflow

1. Définition
2. En mémoire
   - Vue générale
   - Etude de la stack
   - Récapitulatif
3. Smash the stack
   - Stratégie
   - Méthode
4. Exemple
   - Exploitation d'un buffer overflow
5. Questions

Avec la stack

- On sait
  - comment l'espace est alloué
  - comment sont stockés les valeurs
- Plus on copie, plus on monte dans les adresses

Résumé


Stack Growth
Higher Addresses
saved ESI
saved EDI
local variable 3
local variable 2
local variable 1
saved EBP
return address
parameter 1
parameter 2
parameter 3
ESB
EPB
[epb]-4
[epb]+8
[epb]+12
[epb]+16

Justement !

Que se passe-t-il si on continue de copier ?

Buffer overflow

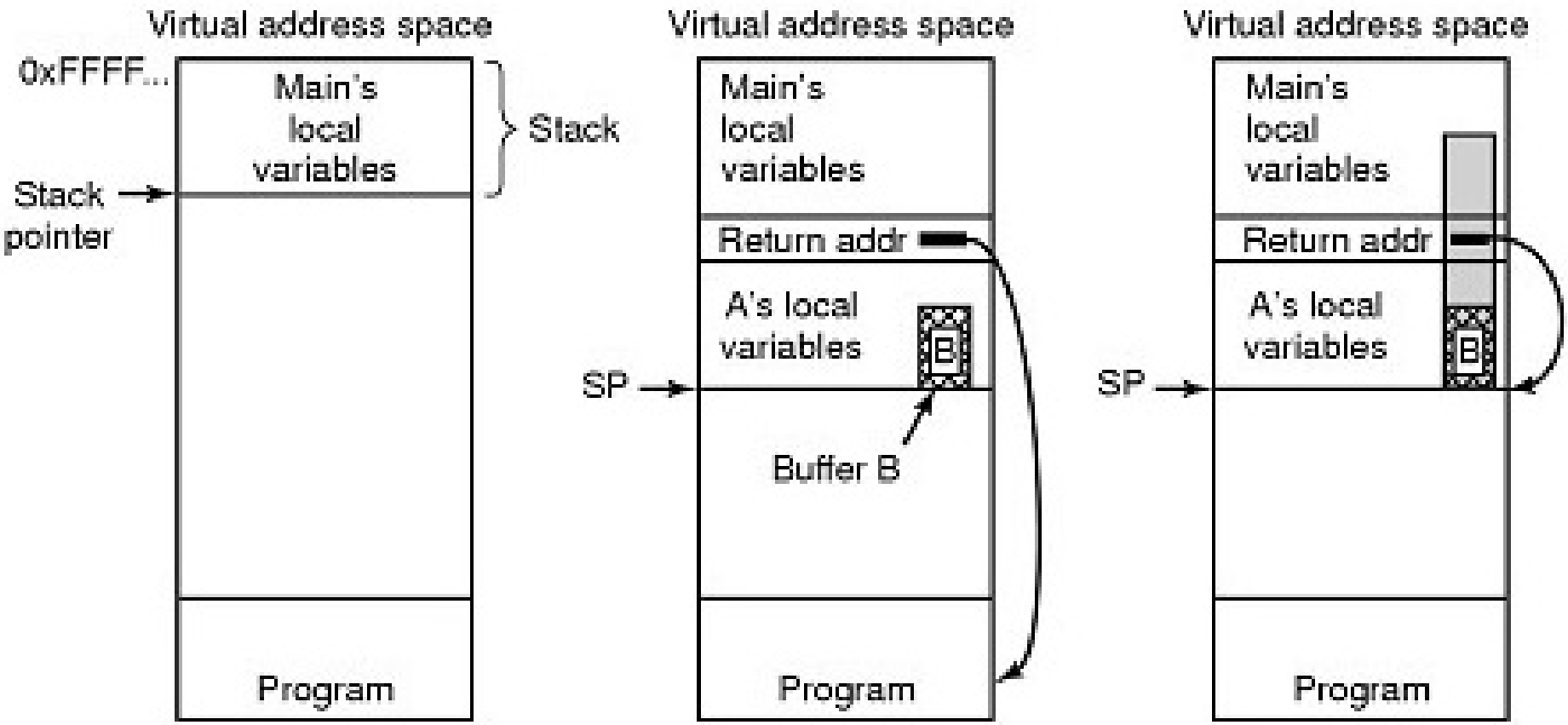
1. Définition
2. En mémoire
   - Vue générale
   - Etude de la stack
   - Récapitulatif
3. Smash the stack
   - Stratégie
   - Méthode
4. Exemple
   - Exploitation d'un buffer overflow
5. Questions

Stratégie

- Plus on écrit, plus les adresses ↗
- L'adresse de retour se trouve plus loin
- Donc, si on dépasse, on **écrasera** l'adresse de retour

$\Rightarrow$Contrôle du flux d'exécution !

Dépasser le buffer



Buffer overflow

1. Définition
2. En mémoire
   - Vue générale
   - Etude de la stack
   - Récapitulatif
3. Smash the stack
   - Stratégie
   - Méthode
4. Exemple
   - Exploitation d'un buffer overflow
5. Questions

Smash the Stack

1. Trouver le buffer overflow
2. Stocker notre shellcode
   - Directement dans le buffer
   - Dans une variable d'environnement
3. Ecraser l'adresse de retour
   - Avec l'adresse du buffer
   - Avec l'adresse de la variable d'env.
4. Poursuivre l'exécution
5. Enjoy \o/

Shellcode

"In computer security, a shellcode is a small piece of code used as the payload in the exploitation of a software vulnerability. It is called "shellcode" because it typically starts a command shell from which the attacker can control the compromised machine. [..] Shellcode is commonly written in machine code." (Wikipedia)

Shellcode

Pour faire simple

- Code machine
- Lance /bin/bash
- Ne peut pas contenir de NUL BYTE (0x00)

Exemple :

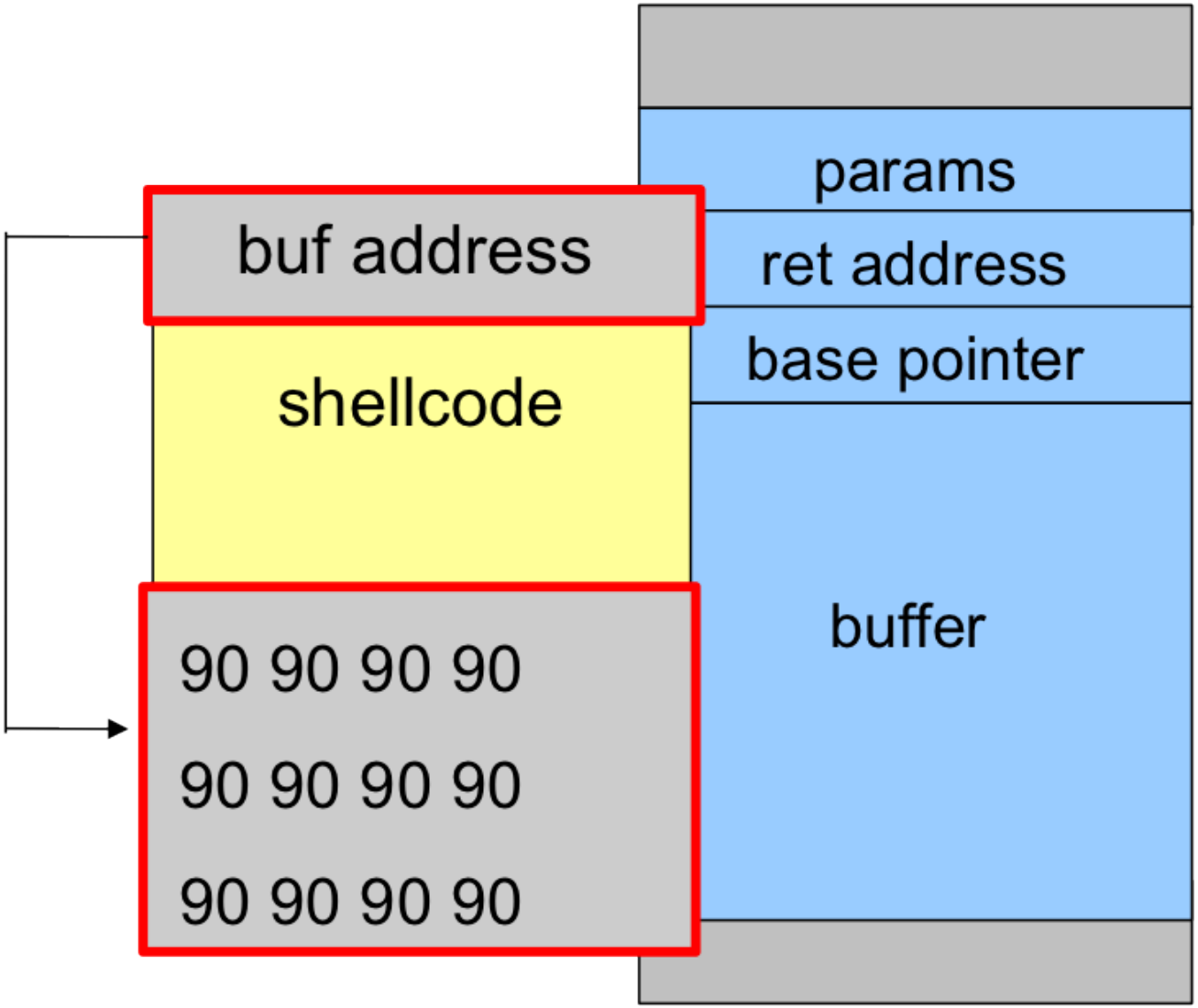
\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e
\x89\xe3\x50 \x89\xe2\x53\x89\xe1\xb0\x0b\xcd\x80

(La création d'un shellcode n'est pas abordée ici)

Contenu de notre buffer

- Contient le shellcode (25 bytes avec l'exemple d'avant)
- Contient l'adresse qui pointe sur le début du shellcode (4 bytes)
- En théorie, le reste n'a pas d'importance
- En pratique, NOP sledding

NOP sledding



Buffer overflow

1. Définition
2. En mémoire
   - Vue générale
   - Etude de la stack
   - Récapitulatif
3. Smash the stack
   - Stratégie
   - Méthode
4. Exemple
   - Exploitation d'un buffer overflow
5. Questions

Code vulnérable

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

int main(int argc, char *argv[]) {
    char buffer[256] = {0};

    if (argc < 2) {
        printf("Usage: %s username\n", argv[0]);
        exit(1);
    }

    /* strcpy is unsafe */
    strcpy(buffer, argv[1]);
    printf("Your username is: %s\n", buffer);

    return 0;
}
```

Problèmes

- Pas de contrôle sur l'input de la part du programmeur
- Taille fixe des buffers
- Utilisation de strcpy

  "BUGS
  If the destination string of a strcpy() is not large enough, then **anything** might happen." (Source : man strcpy)

$\Rightarrow$ Présence d'un buffer overflow !

DEMO

Exploitation d'un buffer overflow (sans protection)

Questions ?

Questions ?