



CHUNGNAM NATIONAL UNIVERSITY



시스템 프로그래밍

강의 7 : 3.12 버퍼 오버플로우

<http://eslab.cnu.ac.kr>

* Some slides are from Original slides of RBE

전달사항

예습 숙제 : 736-745(8.1절)

중간고사 10월 31일 저녁 7시

대비방법

- * 기출문제 – 강의자료게시판
- * 실습과제 이해도
- * 강의 슬라이드속의 연습문제들!!!
- * 강의 자료
- *

10월 28일 강의는 10월 31일로 이동

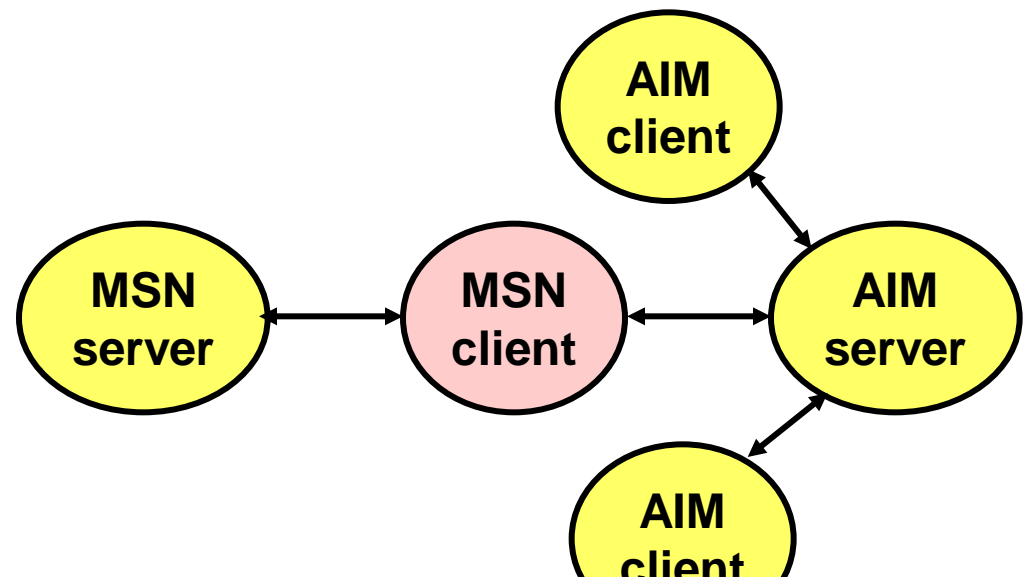
인터넷 웜(Worm)과 IM War

November, 1988, The Morris Worm

- 인터넷 웜은 수천개의 인터넷 호스트를 공격
- 웜 : 스스로 시행되어 다른 기계에 자신을 전파하는 프로그램
- 바이러스 : 다른 프로그램에 자신을 추가하는 프로그램. 비독립적
- How did it happen?

July, 1999

- 마이크로 소프트사는 인터넷 메신저를 출시
- 메신저는 AOL사의 Instant Messaging Service (AIM) 서버를 액세스 할 수 있도록 설계



인터넷 웜(Worm)과 IM War

August 1999

- 갑자기 메신저가 AOL서버를 사용 못하게 됨
- Microsoft 와 AOL 사는 인터넷 메신저 전쟁을 하게 되다:
 - ◆ AOL 은 서버를 교체하여 메신저의 접근을 차단함
 - ◆ 마이크로소프트는 바뀐 서버에 접속할 수 있도록 메신저를 수정
 - ◆ 13차례 반복
- 어떻게 이런 일이 발생했는가?

Internet Worm 와 AOL/Microsoft War 는 모두 *stack buffer overflow* 현상을 이용한 것이다!

- ◆ 많은 Unix functions 는 매개변수의 크기를 체크하지 않는다
- ◆ 버퍼 overflow를 허용한다

스트링 라이브러리 함수

● Unix function gets 의 구현

- ◆ 읽어 들일 수 있는 문자의 갯수를 한정할 수 없는 구조이다

```
/* Get string from stdin */
char *gets(char *dest)
{
    int c = getc();
    char *p = dest;
    while (c != EOF && c != '\n') {
        *p++ = c;
        c = getc();
    }
    *p = '\0';
    return dest;
}
```

● 유사한 다른 Unix 들에서도 같은 문제가 있다

- ◆ strcpy: 임의의 길이의 스트링을 복사
- ◆ scanf, fscanf, sscanf 함수를 %s 와 함께 사용하는 경우

위험한 버퍼 코드

```
/* Echo Line */  
void echo()  
{  
    char buf[4]; /* Way too small! */  
    gets(buf);  
    puts(buf);  
}
```

```
int main()  
{  
    printf("Type a string:");  
    echo();  
    return 0;  
}
```

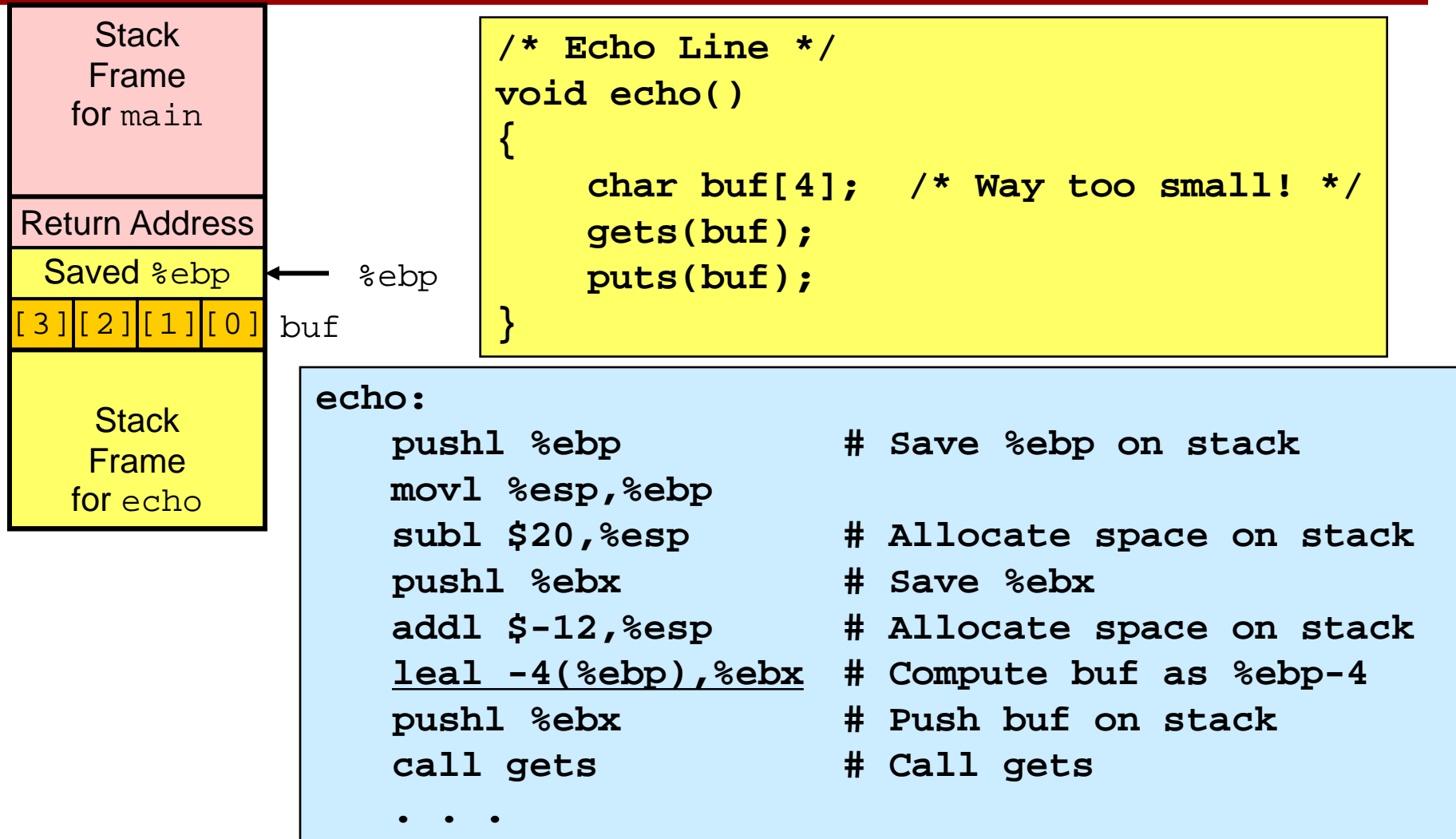
버퍼 오버플로우의 실행

```
unix>./bufdemo  
Type a string:123  
123
```

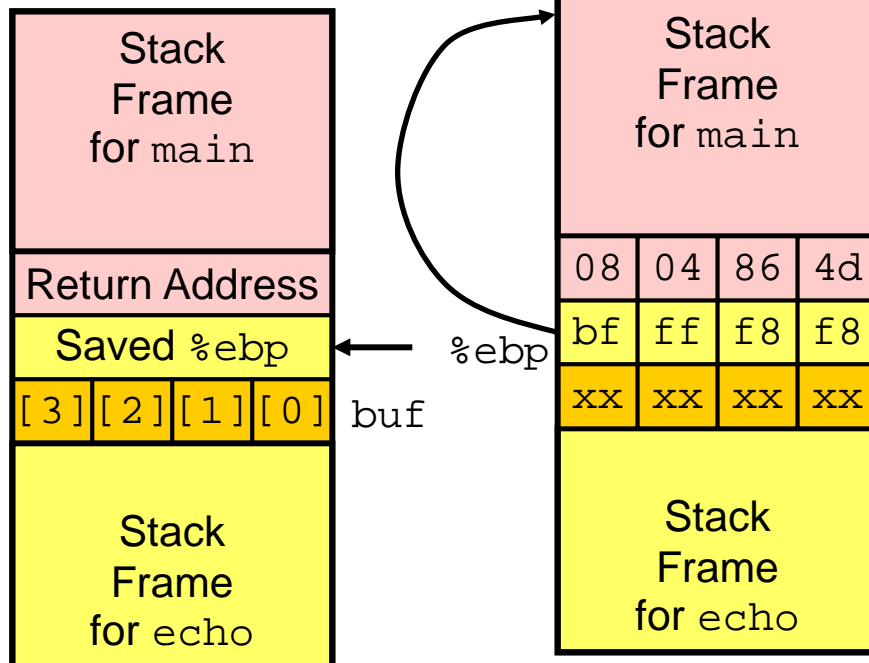
```
unix>./bufdemo  
Type a string:12345678  
Segmentation Fault
```

```
unix>./bufdemo  
Type a string:12345678ABC  
Segmentation Fault
```

버퍼 오버플로우의 스택



버퍼 오버플로우 스택 예제



```

unix> gdb bufdemo
(gdb) break echo
Breakpoint 1 at 0x8048583
(gdb) run
Breakpoint 1, 0x8048583 in echo ()
(gdb) print /x *(unsigned *)$ebp
$1 = 0xbffffff8f8
(gdb) print /x *((unsigned *)$ebp + 1)
$3 = 0x804864d
    
```

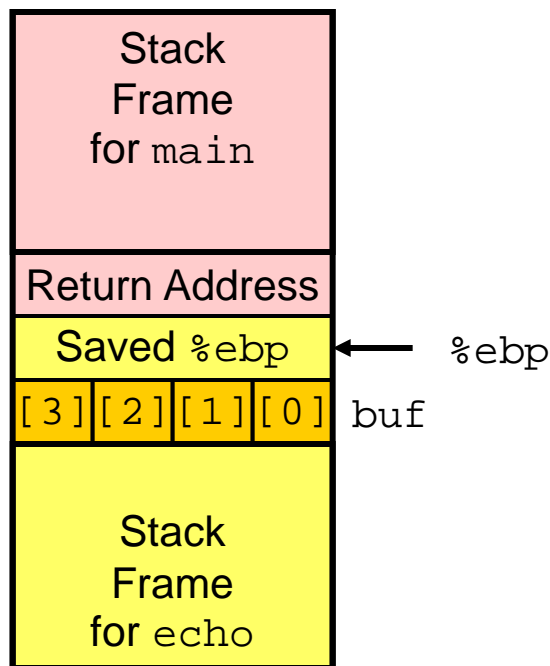
Before call to gets

```

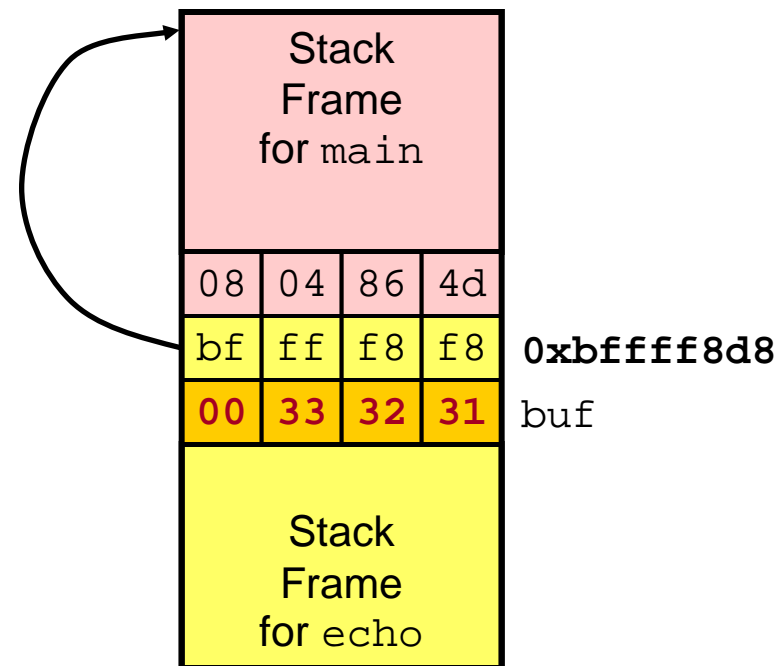
8048648: call 804857c <echo>
804864d: mov 0xffffffe8(%ebp),%ebx # Return Point
    
```

버퍼 오버플로우 예제 1

Before Call to `gets`

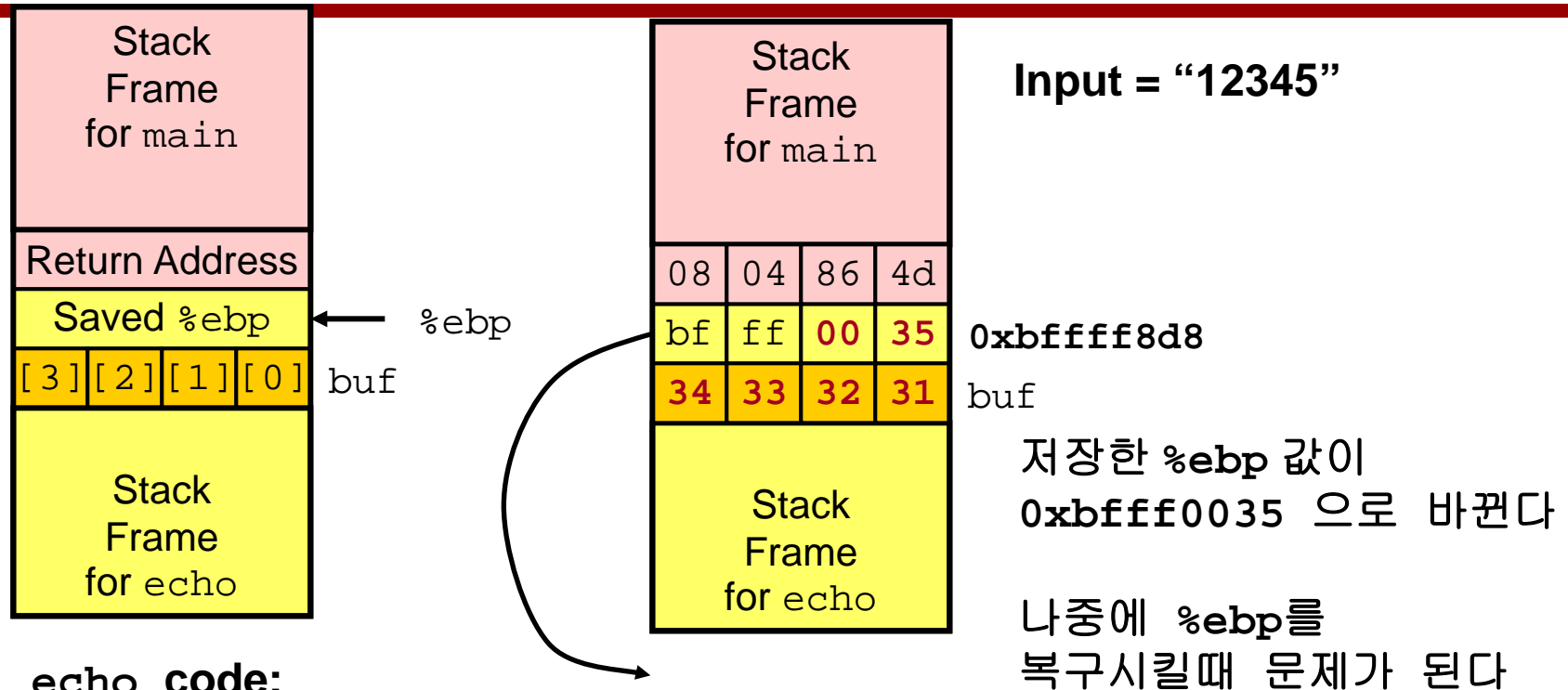


Input = "123"



No Problem

버퍼 오버플로우 예제 2

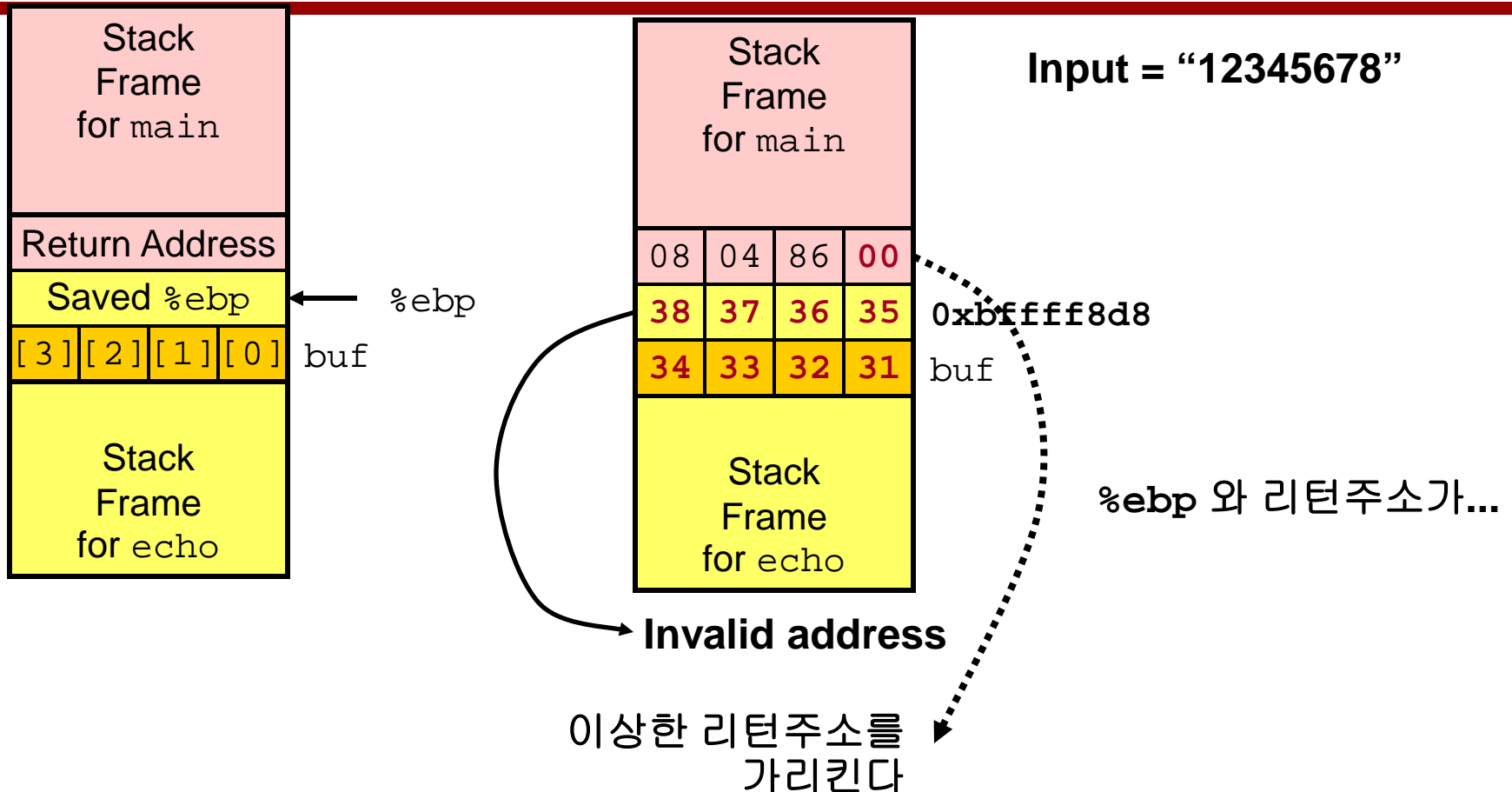


echo code:

```

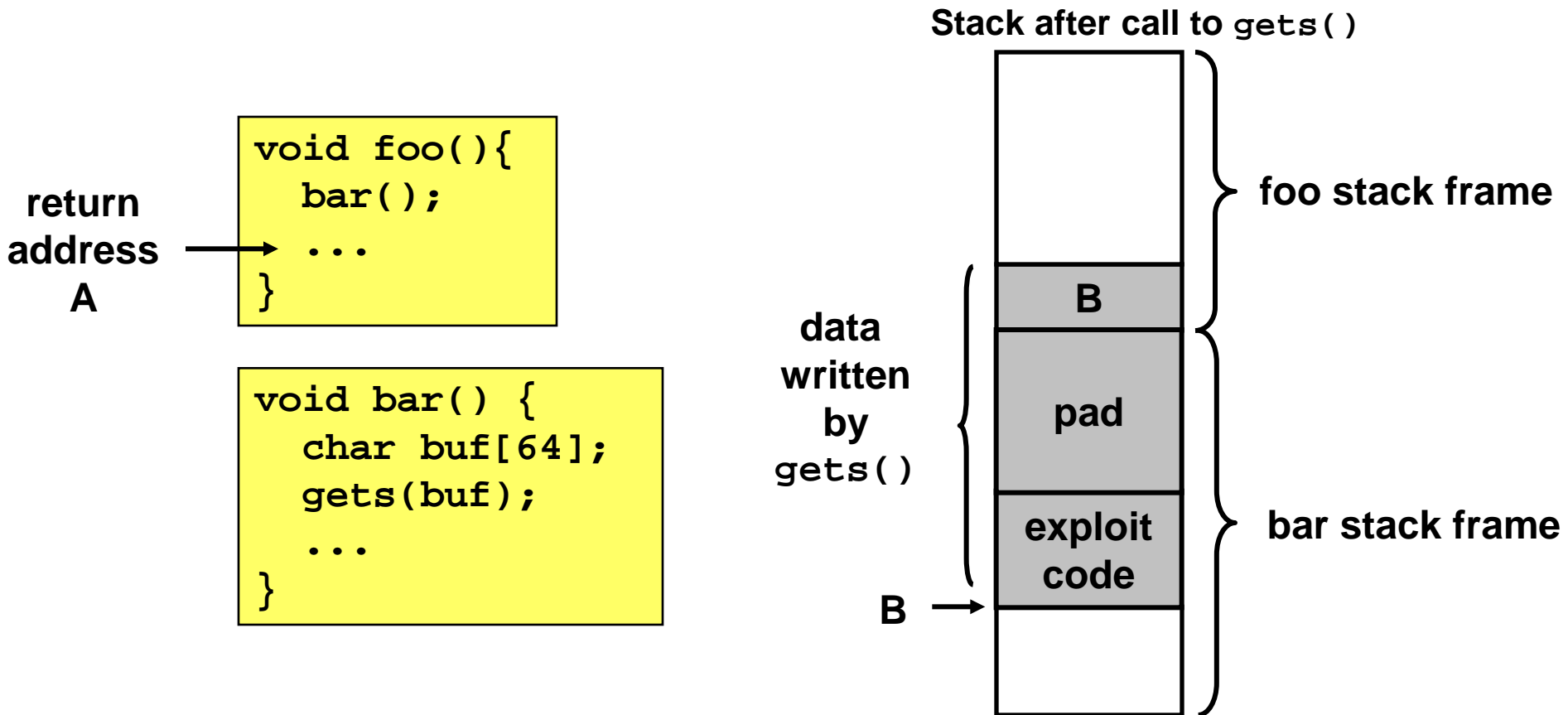
8048592: push    %ebx
8048593: call    80483e4 <_init+0x50> # gets
8048598: mov     0xffffffffe8(%ebp),%ebx
804859b: mov     %ebp,%esp
804859d: pop     %ebp      # %ebp gets set to invalid value
804859e: ret
    
```

버퍼 오버플로우 예제 3



```
8048648: call 804857c <echo>
804864d: mov 0xfffffffffe8(%ebp),%ebx # Return Point
```

버퍼 오버플로우의 사악한 이용



- 입력 스트링에 프로그램 코드를 넣는다
- 리턴 주소 부분에 버퍼의 주소를 써 넣는다
- bar()가 ret을 실행하면, exploit code 부분이 실행된다

버퍼 오버플로우를 이용한 Exploits

버퍼 오버플로우 버그는 원격지 컴퓨터들이 타겟 컴퓨터에서 원하는 프로그램을 실행 시킬 수 있도록 해준다.

Internet worm

- 초기 finger server (fingerd)는 클라이언트가 보내준 매개변수를 읽어 들이기 위해서 `gets()` 를 사용하였다

finger hyungshin@cnu.ac.kr

- Worm 은 다음과 같은 스트링을 fingerd server에 전송하여 공격한다:

◆ *finger "exploit-code padding new-return-address"*

◆ exploit code: 해커와 TCP 직접 연결을 설정하는 root shell 을 타겟 컴퓨터에 실행시킨다.

버퍼 오버플로우를 이용한 Exploits

버퍼 오버플로우 버그는 원격지 컴퓨터들이 타겟 컴퓨터에서 원하는 프로그램을 실행 시킬 수 있도록 해준다.

IM War

- AOL 은 AIM 클라이언트에 존재하는 buffer overflow bug 를 exploit 방식으로 이용하였다.
- exploit code: 4 바이트의 시그니처를 서버로 리턴하도록 함 (AIM 클라이언트의 특정 부분에서 4 바이트를 선택).
- Microsoft사가 시그니처 부분을 찾아내서 메신저를 수정하면, AOL사는 시그니처의 위치를 변경하였음.

Letter from

Date: Wed, 11 Aug 1999 11:30:57 -0700 (PDT)
From: Phil Bucking <philbucking@yahoo.com>
Subject: AOL exploiting buffer overrun bug in their own software!
To: rms@pharlap.com

Mr. Smith,

I am writing you because I have discovered something that I think you might find interesting because you are an Internet security expert with experience in this area. I have also tried to contact AOL but received no response.

I am a developer who has been working on a revolutionary new instant messaging client that should be released later this year.

...

It appears that the AIM client has a buffer overrun bug. By itself this might not be the end of the world, as MS surely has had its share. But AOL is now *exploiting their own buffer overrun bug* to help in its efforts to block MS Instant Messenger.

....

Since you have significant credibility with the press I hope that you can use this information to help inform people that behind AOL's friendly exterior they are nefariously compromising peoples' security.

Sincerely,
Phil Bucking
Founder, Bucking Consulting
philbucking@yahoo.com

나중에 이 편지가 마이크로소프트사에서 보내진 것으로 밝혀졌다!!!

오버플로우 약점을 피하는 방법

```
/* Echo Line */
void echo()
{
    char buf[4]; /* Way too small! */
    fgets(buf, 4, stdin);
    puts(buf);
}
```

스트링의 길이를 제한하는 라이브러리를 사용한다

- fgets instead of gets
- strncpy instead of strcpy
- scanf 를 %s 와 함께 사용하지 않는다
 - ◆ fgets를 사용한다

연습문제 1. 스택 오버플로우

표준 입력에서 한 줄을 읽어서 표준출력으로 써주는 함수 `good_echo()`를 작성하라. 이 함수를 구현할 때 입력하는 문자의 수를 제한해서는 안된다. 라이브러리 함수 `fgets()`를 사용해도 좋지만, 버퍼를 위해 할당한 크기보다 더 많은 문자의 입력이 있는 경우에도 정확히 동작해야 한다. 에러 조건들을 체크해야 하고, 에러 발생시에는 리턴해야 한다. 표준 입출력 함수 문서를 참고해도 좋다.

```
#define BUFSIZE 64
void good_echo ( )
{
    char buf[BUFSIZE];
    int i;
    while (1) {

        }
```

연습문제 1. 스택 오버플로우

```
void good_echo ( )
{
    char buf[BUFSIZE];
    int i;
    while (1) {
        if (!fgets(buf, BUFSIZE, stdin))
            return; /* end of file or error */
        /* print characters in buffer */
        for (i = 0; buf[i] && buf[i] != '\n'; i++)
            if (putchar(buf[i]) == EOF)
                return; /* Error */
        if (buf[i] == '\n') {
            /* Reached terminating newline */
            putchar('\n');
            return;
        }
    }
}
```

요약

3장을 마치며...
앞으로의 일정