버퍼 오버플로우 과제

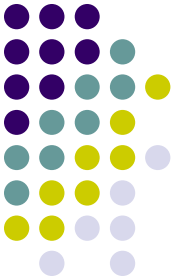
2016.11.01

버퍼오버플로우 과제

- 과제의 목적: BOV(Buffer Overflow)를 이용해서, return address를 바꾸어서, main()의 특정 printf() 문 부터 시작하도록 입력 파일 (binary file)을 준비함.

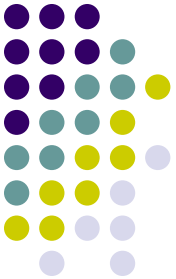
- 준비하기 : 조교가 배정해준 번호에 따라 자신의 홈 디렉토리로 파일을 복사
- cp /home/sys00/bov/bufdemo-nspxx ~
  - (xx에 자신이 할당받은 번호입력)

버퍼오버플로우 과제



- return address 를 바꾸는 방법은: gets()가 –fno-stack-protector option으로 compile 되어 있어서, canary가 포함되어 있지 않음. BOV 약점을 이용해서, gets()에 입력과정에서 리턴주소의 위치에 해당 printf() 문 위치로 점프하도록 주소를 넣어줌으로써, echo()함수 리턴시에 main()의 해당 printf()로 점프하도록 함.
- 실제 수행시에는 아래와 같이 bin(binary, hexa) 파일을 넣음.
- % bufdemo-nsp < bin

binary file 편집 방법



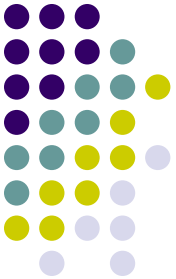
- hexedit를 사용함.

% hexedit bin

- 커서 위치에서 16진수를 그대로 입력함.
- 저장 시에는 CtrI-X를 입력하면->저장 여부 물음 ->Yes 선택 (종료 없이 저장 시 CrtI-w)
- 파일은 없는 경우에는, echo명령으로 binary file을 만들어 낸 후에, hexedit로 편집함.

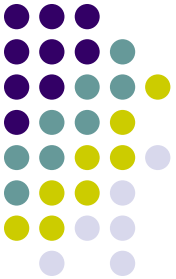
% echo “blablabla” > bin

주의사항



- Binary file 에 주소값을 입력시 개행 문자의 위치에 조심해야한다.
- String 은 개행문자 '\n' 즉, 0A 의 값을 끝으로 인식
- 원치 않은 곳에서 문자열이 끝날 수 있으니 주소 값의 끝이 아닌 곳에 개행 문자를 삽입하지 않도록 주의
- **주소 내에 0A 가 있을 경우, 0A를 포함하지 않는 주소로 점프하도록 변경**

main()의 예제

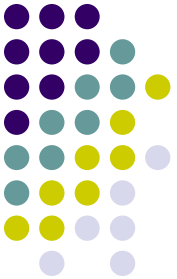


```
int main()
{
    printf("Type a 1 string:"); ← 학번%10 = 1인 학생은 여기로
    printf("Type a 2 string:");  ← 학번%10 = 2인 학생은 여기로
    printf("Type a 3 string:"); ← 학번 %10 =  3인 학생은 여기로
    printf("Type a 4 string:"); ← 학번 %10 = 4인 학생은 여기로
    printf("Type a 5 string:"); ← 학번 %10 = 5인 학생은 여기로
    .   .   .
    printf("Type a 10 string:"); ← 학번 %10 = 0인 학생은 여기로
    call_echo();
    return 0;
}
```

call_echo(), echo()

```
void call_echo() {
echo();
}
void echo() {
char buf2[MAXLEN];
char buf[8];
gets(buf, buf2);
puts(buf);
}
```

예:
학번%10=0인 학생의 수행 결과...



%bufdemo-nsp < bin

Type a string: ... Type a 10 string: aslkjslakjdkajasdlkjdsa ...

Type a 10 string: ...

제출할 사항(자신이 받은 bufdemo-nspxx 파일에 대해서)

1. binary file (bufdemo-nspxx의 입력 파일) - 제출파일
2. echo() 함수의 return address ?
3. echo() 함수의 return address의 저장 위치 (16진수로,stack상) ?
4. echo() 함수에서 buf와 buf2의 시작 주소 (16진수로,stack상) ?

1은 제출물로, 2-4는 문서로 작성해서, 1과 함께 zip 파일로 제출함.