

Мета

Дослідити можливі вразливості переповнення буферу.

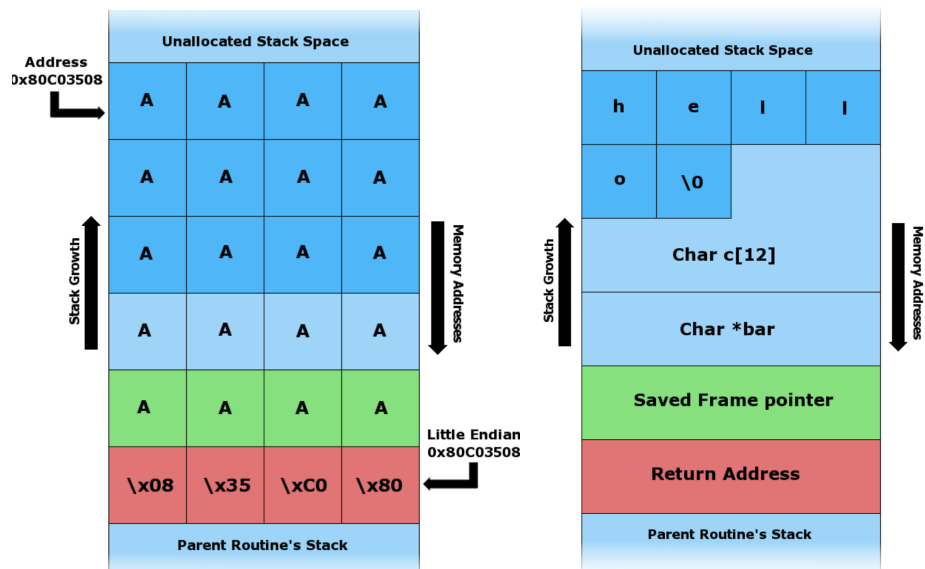
Завдання:

Взяти за основу наведений текст програми.

ХІД РОБОТИ

В галузі комп'ютерної безпеки і програмування, переповнення буфера (англ. buffer overflow або англ. buffer overrun), це явище, при якому програма, під час запису даних в буфер, перезаписує дані за межами буфера. Це може викликати несподівану поведінку, включно з помилками доступу до даних, невірними результатами, збоєм програми або дірою в системі безпеки. Переповнення буфера може бути викликане недостатньою перевіркою вхідних даних. Воно є базою для багатьох уразливостей в програмних продуктах і може бути злонамірено використане. Додаткова перевірка може запобігти переповненню буфера, хоча така перевірка відіб'ється на швидкодії програми. Мови програмування, зазвичай згадувані у зв'язку з переповненням буфера, це здебільшого С та С++. Вони не мають вбудованого механізму проти доступу або перезаписування даних у будь-якій частині пам'яті і не провадять автоматичної перевірки даних, які записують в масив, на вихід за межі масиву.

В програмах, переповнення буфера стеку трапляється, коли програма пише за адресами в програмному стеку виклику поза призначеною структурою даних; це зазвичай буфер фіксованої довжини. Баг переповнення буфера стеку трапляється, коли програма пише більше даних в буфер розміщений в стеку, ніж було фактично виділено місця для буфера. Це майже завжди призводить до псування прилеглих даних в стеку, і в разі якщо переповнення було зроблено помилково, часто призводить до краху програми або некоректної роботи. Цей тип переповнення є одним з випадків загальнішого класу багів програмування, знаних як переповнення буфера.



Якщо атакована програма виконується з спеціальними привілеями, або приймає дані з недовірених хостів мережі (напр. вебсерверів), тоді баг є потенційною вразливістю безпеки. Якщо стековий буфер заливий даними, які надійшли від недовіреного користувача, тоді цей користувач може пошкодити стек в такий спосіб, що в стеку опиняється виконуваний код, інжектований ним, відтак він отримує управління процесом. Це один з найстаріших і найнадійніших методів для зломисників отримати неавторизований доступ до комп'ютера.

Беремо код вихідний програми

```
bool isPasswordOk() {
    char password[12];
    gets(password);
    return 0 == strcmp(password, "goodpass");
}

int main(int argc, char* argv[]) {
    bool status;
    puts("Enter password: ");
    status = isPasswordOk();
    if (!status) {
        puts("Access denied");
        exit(-1);
    }
    puts("Passed");
    return 0;
}
```

Далі відкриваємо дизасембльований код

00401095	EB 05	jmp lab11.40109c	
00401097	18C0	sbb eax,eax	
00401099	83C8 01	or eax,1	
0040109C	85C0	test eax,eax	
0040109E	74 12	je lab11.4010B2	main.cpp:18
004010A0	68 1C214000	push lab11.40211c	main.cpp:20, 40211c:"Access
004010A5	FFD6	call esi	
004010A7	83C4 04	add esp,4	main.cpp:21
004010AA	6A FF	push FFFFFFFF	
004010AC	FF15 5C204000	call dword ptr ds:[<&exit>]	main.cpp:23, 40212c:"Passed"
004010B2	68 2C214000	push lab11.40212c	
004010B7	FFD6	call esi	main.cpp:25
004010B9	8B4D FC	mov ecx,dword ptr ss:[ebp-4]	
004010BC	83C4 04	add esp,4	
004010BF	33CD	xor ecx,ebp	
004010C1	33C0	xor eax,eax	
004010C3	5E	pop esi	
004010C4	E8 04000000	call <lab11.@@security_check_cookie@4>	

Рисунок 1 - Дизасемблований код

0019FEF4	00000000		
0019FEF8	0019FF0C	&"123456789012345678"	
0019FEFC	77433EA0	ucrtbase.77433EA0	
0019FF00	0019FF28		
0019FF04	004010B2	возврат к lab11.main+72 из ???	
0019FF08	004020F8	lab11.004020F8	
0019FF0C	0019FF18	"123456789012345678"	
0019FF10	00402108	"Enter password: "	
0019FF14	774900E0	ucrtbase.774900E0	
0019FF18	34333231		
004010B2	68 2C214000	push lab11.40212c	main.cpp:23, 40212c:"Passed"
004010B7	FFD6	call esi	
004010B9	8B4D FC	mov ecx,dword ptr ss:[ebp-4]	main.cpp:25
004010BC	83C4 04	add esp,4	
004010BF	33CD	xor ecx,ebp	
004010C1	33C0	xor eax,eax	
004010C3	5E	pop esi	
004010C4	E8 04000000	call <lab11.@@security_check_cookie@4>	
004010C9	8BE5	mov esp,ebp	
004010CB	5D	pop ebp	
004010CC	C3	ret	
004010CD	3B0D 04304000	cmp ecx,dword ptr ds:[<__security_cookie@4>]	secchk.c:53, 00403004:"!A"ha
004010D3	F2:75 02	bnd jne <lab11.failure>	secchk.c:57
004010D6	F2:C3	bnd ret	secchk.c:58
004010D8	F2:E9 79020000	bnd jmp <lab11.____report_gsfailure>	secchk.c:60
004010DE	56	push esi	exe_common.inl:147

Рисунок 2 – Адреса повернення

Після заміни адреси повернення: програма попадає в місце, яке виконується тільки при вірному значенні пароля.

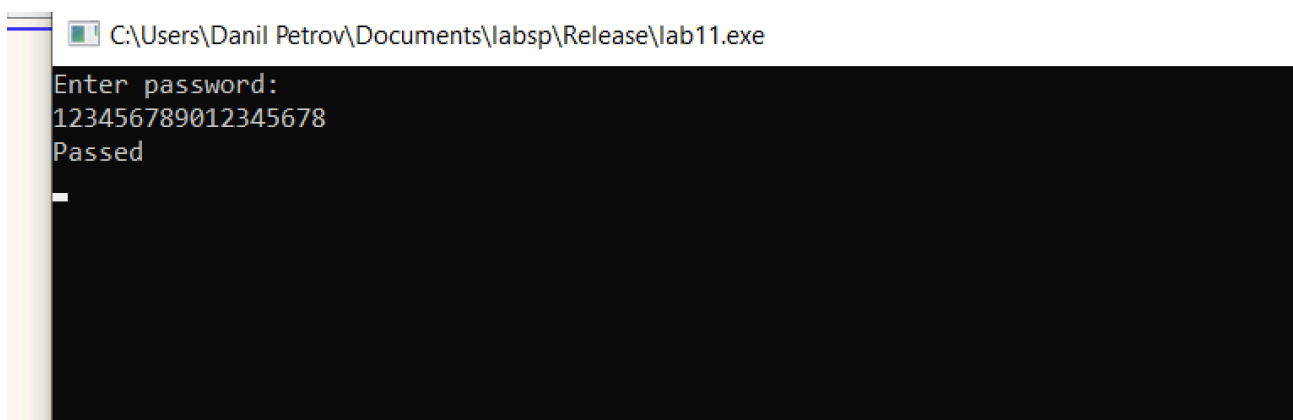


Рисунок – Ввід паролю

Висновок: Дослідили можливі вразливості переповнення буферу.