

Jewellery Store Management System has buffer overflow in searchitem function

supplier

<https://code-projects.org/jewellery-store-management-system-in-c-programming-with-source-code/>

Vulnerability file

searchitem function

describe

There is a stack overflow vulnerability in searchitem function of Jewellery Store Management System. it cause ddos and rce.

Code analysis

In search function,get unlimited data saved in str2 variable, resulting in buffer overflow vulnerability.

```
51 }
52 }
53 if ( findstaff != 116 )
54     printf("\nNo Record Found");
55     gotoxy(20164, 17164);
56     printf("Try another search?(Y/N)");
57     if ( getch() == 121 )
58         searchitem();
59     else
60         mainmenu();
61 }
62 else if ( v0 == 50 )
63 {
64     system("cls");
65     gotoxy(25164, 4164);
66     printf(&byte_406488);
67     gotoxy(20164, 5164);
68     printf("Enter Item's Name:");
69     scanf("%s", str2);
70     v0 = 0;
71     while ( fread(&a, 0x70ui64, 1ui64, fp) == 1 )
72     {
73         if ( !strcmp((const char *)&a + 24, str2) )
74         {
75             gotoxy(20164, (unsigned int)(v4 + 7));
76             gotoxy(20164, (unsigned int)(v4 + 8));
77             printf("Item Code:%d", a);
78             gotoxy(20164, (unsigned int)(v4 + 10));
79             printf("Name:%s", (const char *)&a + 24);
80             gotoxy(20164, (unsigned int)(v4 + 11));
81             printf("Material:%s", (const char *)&a + 44);
82             gotoxy(20164, (unsigned int)(v4 + 12));
83         }
84     }
85 }
```

000016DF searchitem:69 (4020DF)

Output

```
4024E6: using guessed type __int64 edititem(void);
402B03: using guessed type __int64 t(void);
402C31: using guessed type __int64 calculatebill(void);
401530: using guessed type __int64 __fastcall gotoxy(_QWORD, _QWORD);
401530: using guessed type __int64 __fastcall gotoxy(_QWORD, _QWORD);
```

POC

enter pass

pass

```

WELCOME
To
膊膊膊 Jewellery Store Management System 膊膊膊

Enter Password:****|
pass

```

[illegible]

2. Search By Name

2

[illegible]

we can see EXCEPTION_ACCESS_VIOLATION.

