

PICOC Null Pointer Dereference Denial of Service

PICOC Suffers from a Denial of Service (CWE476) vulnerability as a result of a Null Pointer Dereference. Any project or library that uses Picoc also suffers from this issue. An example of this would be picoc-js (<https://www.npmjs.com/package/picoc-js>). As a result PICOC will immediately segfault.

Reproduction Steps

1. Create a file to be executed by the PICOC interpreter

```
$ touch vulncode
```

2. Add the following code to the file:

```
printf("Before Crash\n");  
**4%;  
printf("This code won't execute because of the crash\n");
```

3. Execute PICOC against the file:

```
$ ./picoc -s vulncode
```

4. You will receive a segfault and the program will crash. This is a result of a null pointer dereference that is not caught or handled by the interpreter. The vulnerable line of code can be seen below:

```
**4%;
```

Solution

Adding a few if statements that verify the pointer is not NULL before usage will solve this problem. You can find more information about this here:

https://owasp.org/www-community/vulnerabilities/Null_Dereference

To upload designs, you'll need to enable LFS and have an admin enable hashed storage. [More information](#)

Tasks  0

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items  0

Link issues together to show that they're related or that one is blocking others. [Learn more.](#)

Activity



Halcy0nic @Halcy0nic · 4 months ago

Author

GDB Trace:

```
Starting program: /home/hali/projects/fuzzing/pic0n/pic0n vuln/crash1
Program received signal SIGSEGV, Segmentation fault.
variableDereferencePointer (PointerVal=0x020200, DereferVal=0x7fffffff580, DereferOffset=0x7fffffff58c, DereferType=0x7fffffff578, DereferIsValue=0x7fffffff580) at variable.c:519
519      DereferType = PointerValue_Type_FromType(
LEGEND: STACK | HEAP | CODE | DATA | ROM | XMM | REGISTER | DISASM |
[ REGISTER ]
RAX 0x0
RDX 0x0
RCX 0x7fffffff570 ← 0x0
RDX 0x7fffffff58c ← 0x0202000000000000
RDI 0x7fffffff58c ← 0x0
R12 0x7fffffff580 ← 0x0
R8 0x7fffffff580 ← 0x1
R9 0x0
R10 0xffffffffffff90d
R11 0x20
R12 0x0078 [__libc_start] ← aux ebp, ebp
R13 0x0
R14 0x0
R15 0x0
RSP 0x7fffffff540 → 0x7fffffff540 → 0x7fffffff620 → 0x7fffffff680 → 0x7fffffff700 ← ...
RBP 0x7fffffff540 → 0x7fffffff540 → 0x7fffffff620 → 0x7fffffff680 → 0x7fffffff700 ← ...
RIP 0x413f48 <variableDereferencePointer+10> ← mov rax, qword ptr [rax + 0x10]
[ DISASM ]
0x413f48 <variableDereferencePointer+56> mov rax, qword ptr [rax + 0x10]
0x413f4A <variableDereferencePointer+60> mov rax, qword ptr [rbp + 0x20]
0x413f4B <variableDereferencePointer+64> mov qword ptr [rax], rcx
0x413f4D <variableDereferencePointer+67> cmp qword ptr [rbp + 0x10], 0
0x413f4E <variableDereferencePointer+72> je VariableDereferencePointer+84 <variableDereferencePointer+8>
+
0x413f4C <variableDereferencePointer+84> cmp qword ptr [rbp + 0x20], 0
0x413f4F <variableDereferencePointer+89> je VariableDereferencePointer+101 <variableDereferencePointer+10>
+
0x413f4D <variableDereferencePointer+89> mov rax, qword ptr [rbp + 0]
0x413f4E <variableDereferencePointer+105> mov rax, qword ptr [rax + 0]
0x413f4D <variableDereferencePointer+109> mov rax, qword ptr [rax]
0x413f4E <variableDereferencePointer+112> mov rbp
```

Please [register](#) or [sign in](#) to reply