

Relatório Warm-up T&C

Joaquim Lucena, Christian Bobsin

Testes de Classes de Equivalência

Nome do Teste : Nome do teste utilizado na ferramenta Unity. Junto uma breve explicação

Número do Teste: Número representando a ordem do teste dentro do conjunto dele na ferramenta da Unity.

valid_s():

Número do Teste	Nome do Teste	Casos de Teste
4	validS - teste da classe válida	[{'D'}, {1}]
6	validS - teste da classe inválida	[{' '}, {0}]

valid_f():

Número do Teste	Nome do Teste	Casos de Teste
2	validF - teste da classe válida	[{' '}, {0}]
4	validF - teste da classe inválida	[{'5'}, {1}]

identifier():

Número do Teste	Nome do Teste	Casos de Teste
1	equivalencia - teste do ID com todos os caracteres corretos	[{"T1C"}, {0}]
2	equivalencia - teste do ID com algum caractere inválido	[{"D!C8"}, {1}]

3	equivalencia - teste do ID com primeiro caractere inválido	[{"0AC9"},{1}]
---	--	----------------

Testes de Valor Limite

valid_s():

Número do Teste	Nome do Teste	Casos de Teste
1	validS - menor valor dentro do limite	[{'A'},{1}]
2	validS - maior valor mais próximo do limite inferior	[{'A' - 1},{0}]
3	validS - maior valor dentro do limite	[{'z'},{1}]
5	validS - menor valor mais próximo do limite superior	[{'z' + 1},{0}]

valid_f():

Número do Teste	Nome do Teste	Casos de Teste
1	validF - menor valor dentro do limite	[{'0'},{1}]
3	validF - maior valor mais próximo do limite inferior	[{'z'},{1}]
5	validF - maior valor dentro do limite	[{'0' - 1},{0}]
6	validF - menor valor mais próximo	[{'z' + 1},{0}]

	do limite superior	
--	--------------------	--

identifier():

Número do Teste	Nome do Teste	Casos de Teste
1	limiteTamanho - Apenas 1 caractere	[{"A\n"}, {0}]
2	limiteTamanho - 6 caracteres	[{"A00000\n"}, {0}]
3	limiteTamanho - 6 + 1 caracteres	[{"A000000\n"}, {1}]
4	limiteTamanho - 0 caracteres	[{"\n"}, {1}]

Número do Teste	Nome do Teste	Casos de Teste
1	limiteValor - último caractere com valor '0' - 1	[{"A0000/\n"}, {1}]
2	limiteValor - Último caractere com valor 'z' + 1	[{"zzzz\n"}, {1}]
3	limiteValor - ('z'+1), teste no primeiro caractere	[{"{\n"}, {1}]
4	limiteValor - ('A'-1), teste no primeiro caractere	[{"@\n"}, {1}]

Cobertura do Gcov

```
File src/identifier.c Lines executed:100.00% of 26 Branches executed:100.00% of 32 Taken at least once:96.88% of 32 Calls executed:100.00% of 5
```

Observações

- Na hora de fazermos a mudança para passarmos uma string como argumento ao invés de usarmos um fgets, a nossa implementação foi feita com um erro de acesso

à memória. Felizmente, o addrsanitizer conseguiu pegar o erro. Abaixo o print do terminal quando o erro acontece. Este erro pode ser “evitado” comentando o teste 4 do limiteTamanho(o teste onde passamos um identificador vazio, apenas com \n) caso seja necessário olhar a cobertura do gcov ao vivo.

```
==1058==ERROR: AddressSanitizer: global-buffer-overflow on address 0x7fc5572171a2 at pc 0x7fc557212ac0 bp 0x7fffc1061e70 sp 0x7fffc1061e60
READ of size 1 at 0x7fc5572171a2 thread T0
#0 0x7fc557212abf in identifier src/identifier.c:39
#1 0x7fc557212369 in TEST_Identifier_limiteTamanho_test/TestID.c:51
#2 0x7fc557210583 in UnityTestRunner ../../extras/fixture/src/unity_fixture.c:105
#3 0x7fc55721227b in TEST_Identifier_limiteTamanho_run test/TestID.c:46
#4 0x7fc557212687 in TEST_Identifier_GROUP_RUNNER test/test_runners/TestID_Runner.c:8
#5 0x7fc55721273a in RunAllTests test/test_runners/all_tests.c:5
#6 0x7fc55721014b in UnityMain ../../extras/fixture/src/unity_fixture.c:41
#7 0x7fc557212788 in main test/test_runners/all_tests.c:10
#8 0x7fc555a61bf6 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21bf6)
#9 0x7fc55720a2c9 in _start (/mnt/f/Trabalhos/20211/TEC/TEC_Travis/code/identifier/all_tests.out+0xa2c9)

0x7fc5572171a2 is located 62 bytes to the left of global variable '*.LC17' defined in 'test/TestID.c' (0x7fc5572171e0) of size 12
'*.LC17' is ascii string 'limiteValor'
0x7fc5572171a2 is located 0 bytes to the right of global variable '*.LC16' defined in 'test/TestID.c' (0x7fc5572171a0) of size 2
'*.LC16' is ascii string '
',
SUMMARY: AddressSanitizer: global-buffer-overflow src/identifier.c:39 in identifier
Shadow bytes around the buggy address:
 0x0ff92ae3ade0: f9 f9 f9 00 05 f9 f9 f9 f9 f9 00 00 00 07
 0x0ff92ae3adf0: f9 f9 f9 03 f9 f9 f9 f9 f9 05 f9 f9 f9
 0x0ff92ae3ae00: f9 f9 f9 06 f9 f9 f9 f9 f9 06 f9 f9 f9
 0x0ff92ae3ae10: f9 f9 f9 06 f9 f9 f9 f9 f9 00 00 00 00
 0x0ff92ae3ae20: f9 f9 f9 03 f9 f9 f9 f9 f9 00 01 f9 f9
 0x0ff92ae3ae30: f9 f9 f9 02 f9 f9 f9 f9 f9 00 04 f9 f9
 0x0ff92ae3ae40: f9 f9 f9 00 00 00 06 f9 f9 f9 07 f9 f9
 0x0ff92ae3ae50: f9 f9 f9 07 f9 f9 f9 f9 f9 07 f9 f9 f9
 0x0ff92ae3ae60: f9 f9 f9 03 f9 f9 f9 f9 f9 03 f9 f9 f9
 0x0ff92ae3ae70: f9 f9 f9 00 00 00 00 00 00 02 f9 f9 f9
 0x0ff92ae3ae80: f9 f9 f9 00 00 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable: 00
Partially addressable: 01 02 03 04 05 06 07
Heap left redzone: fa
Freed heap region: fd
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack after return: f5
Stack use after scope: f8
Global redzone: f9
Global init order: f6
Poisoned by user: f7
Container overflow: fc
Array cookie: ac
Intra object redzone: bb
ASan internal: fe
Left alloca redzone: ca
Right alloca redzone: cb
==1058==ABORTING
Makefile:75: recipe for target 'addrsanitizer' failed
make: *** [addrsanitizer] Error 1
```