

Analysis Results

itc-benchmarks-test.git

Report Date

02/26/2020 17:24:23

Report Author

marco.macala@gmail.com

Classification Method

By severity

Product Version

3.4.0

TABLE OF CONTENTS

PROJECT INFORMATION	3
Security Level Dynamics	3
Vulnerability Dynamics	3
Scan History	4
Scan Information 02/26/2020 17:00:46	5
Scan Statistics	5
Language Statistics	6
Vulnerability Table	7
Detailed Results	51
Scan Settings	1121
Export Settings	1123

PROJECT INFORMATION

Project	itc-benchmarks-test.git
UUID	e38fb5d7-b202-4248-a70d-38fc667e98a6
Go To Results In	SmartDec Scanner
Source code	
	https://github.com/mmacala/itc-benchmarks-test.git



Security Level Dynamics

App score calculated on a scale of 0 to 5. Score is calculated based on the number of critical and medium level vulnerabilities. The impact of critical vulnerabilities is greater than that of medium level vulnerabilities, and does not take into account the amount of code. Medium level vulnerabilities are taken into account based on their frequency and total number of source code lines.

Vulnerability Dynamics

Vulnerabilities are divided into three categories: critical, intermediate level and low level.

1. **Critical vulnerabilities** are likely to compromise sensitive data and system integrity.
2. **Medium level vulnerabilities** are less likely to compromise confidential data and system integrity, or are less serious security breaches.
3. **Low level vulnerabilities** signal a violation of good programming practices.

First of all, pay attention to vulnerabilities of critical and medium levels.

Scan History

Number	Date and Time	Status	Languages	Lines of Code	Number of Vulnerabilities				Score
					Critical	Medium	Low	Total	
1/1	02/26/2020 17:00:46	Scan completed	C/C++, Objective-C	41 599	5	733	232	970	2.0/5.0

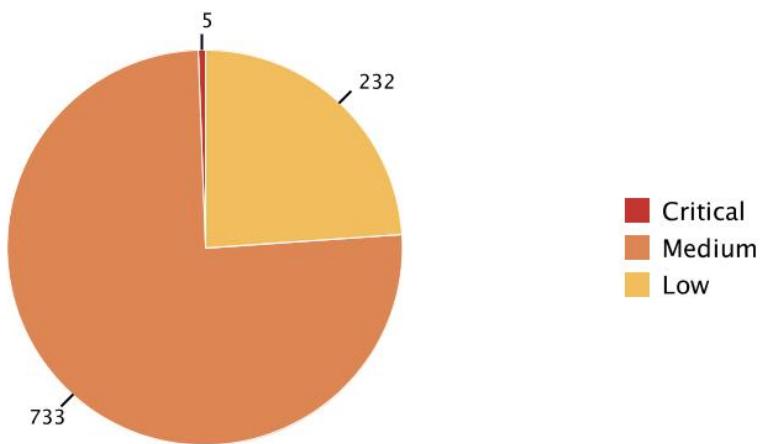
SCAN INFORMATION

02/26/2020 17:00:46

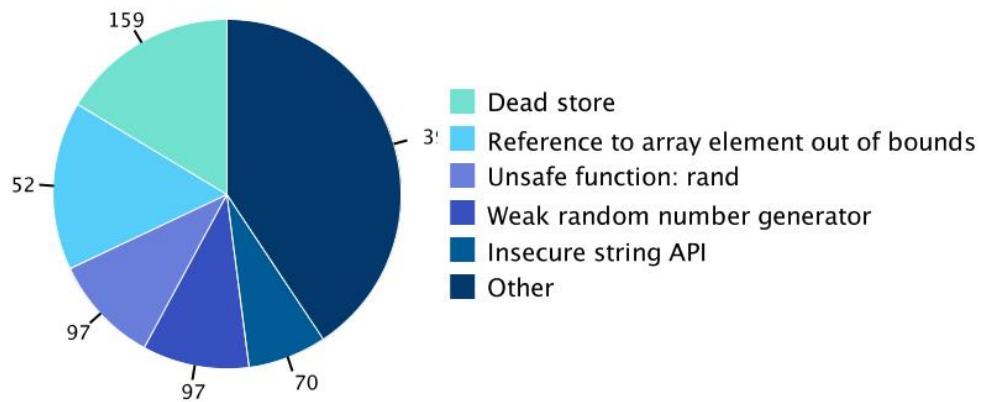
Scan Statistics

Status	Scan completed							
Score	2.0/5.0							
Duration	0:04:19							
Lines of Code	41 599							
Vulnerabilities	Critical	5	Medium	733	Low	232	Total	970

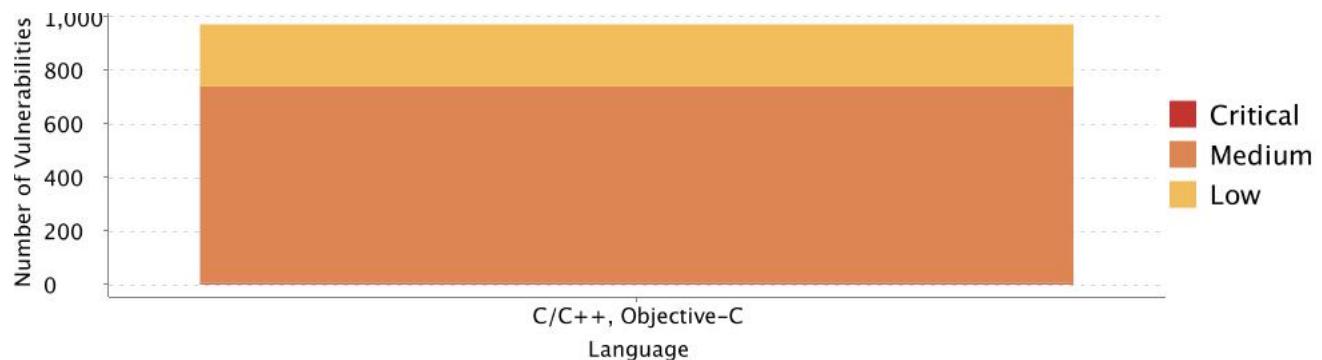
Found Vulnerabilities



Vulnerability Types



Language Statistics



Language	Status	Duration	Lines of Code	Number of Vulnerabilities			
				Critical	Medium	Low	Total
C/C++, Objective-C	complete	0:04:19	41 585	5	733	232	970

Vulnerability Table

Vulnerabilities are displayed depending on export settings.

Critical vulnerabilities		5*
Buffer overflow	C/C++	2
01.w_Defects/st_underrun.c:204		Not processed
02.wo_Defects/st_underrun.c:213		Not processed
Dangerous pointer manipulation	C/C++	3
01.w_Defects/ptr_subtraction.c:22		Not processed
01.w_Defects/ptr_subtraction.c:35		Not processed
02.wo_Defects/ptr_subtraction.c:23		Not processed
Medium-level vulnerabilities		733*
Division by zero	C/C++	13
01.w_Defects/zero_division.c:23		Not processed
01.w_Defects/zero_division.c:34		Not processed
01.w_Defects/zero_division.c:47		Not processed
01.w_Defects/zero_division.c:78		Not processed
01.w_Defects/zero_division.c:118		Not processed
01.w_Defects/zero_division.c:141		Not processed
01.w_Defects/zero_division.c:166		Not processed
01.w_Defects/zero_division.c:178		Not processed
01.w_Defects/zero_division.c:195		Not processed
01.w_Defects/zero_division.c:206		Not processed
01.w_Defects/zero_division.c:225		Not processed
01.w_Defects/zero_division.c:252		Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Division by zero

C/C++

03.w_Defects_Cpp/improper_error_handling.cpp:22

Not processed

Double free

C/C++

12

01.w_Defects/double_free.c:22

Not processed

01.w_Defects/double_free.c:43

Not processed

01.w_Defects/double_free.c:64

Not processed

01.w_Defects/double_free.c:87

Not processed

01.w_Defects/double_free.c:101

Not processed

01.w_Defects/double_free.c:115

Not processed

01.w_Defects/double_free.c:131

Not processed

01.w_Defects/double_free.c:149

Not processed

01.w_Defects/double_free.c:168

Not processed

01.w_Defects/double_free.c:187

Not processed

01.w_Defects/double_free.c:204

Not processed

01.w_Defects/double_free.c:222

Not processed

Function calling with invalid null argument

C/C++

9

01.w_Defects/free_null_pointer.c:109

Not processed

01.w_Defects/free_null_pointer.c:146

Not processed

01.w_Defects/free_null_pointer.c:241

Not processed

01.w_Defects/memory_leak.c:424

Not processed

01.w_Defects/null_pointer.c:238

Not processed

02.wo_Defects/free_null_pointer.c:119

Not processed

02.wo_Defects/free_null_pointer.c:249

Not processed

02.wo_Defects/memory_leak.c:431

Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Function calling with invalid null argument

C/C++

02.wo_Defects/null_pointer.c:259

Not processed

Identical expressions

C/C++

1

01.w_Defects/pow_related_errors.c:441

Not processed

Incompatible allocated type

C/C++

2

01.w_Defects/uninit_pointer.c:215

Not processed

02.wo_Defects/uninit_pointer.c:227

Not processed

Insecure string API

C/C++

70

01.w_Defects/free_null_pointer.c:109

Not processed

01.w_Defects/free_null_pointer.c:146

Not processed

01.w_Defects/free_null_pointer.c:241

Not processed

01.w_Defects/free_null_pointer.c:275

Not processed

01.w_Defects/free_null_pointer.c:320

Not processed

01.w_Defects/invalid_memory_access.c:102

Not processed

01.w_Defects/invalid_memory_access.c:205

Not processed

01.w_Defects/invalid_memory_access.c:210

Not processed

01.w_Defects/invalid_memory_access.c:560

Not processed

01.w_Defects/invalid_memory_access.c:568

Not processed

01.w_Defects/invalid_memory_access.c:611

Not processed

01.w_Defects/invalid_memory_access.c:622

Not processed

01.w_Defects/memory_allocation_failure.c:495

Not processed

01.w_Defects/memory_allocation_failure.c:577

Not processed

01.w_Defects/memory_leak.c:73

Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Insecure string API

C/C++

 01.w_Defects/memory_leak.c:96	Not processed
 01.w_Defects/memory_leak.c:270	Not processed
 01.w_Defects/memory_leak.c:402	Not processed
 01.w_Defects/memory_leak.c:424	Not processed
 01.w_Defects/memory_leak.c:517	Not processed
 01.w_Defects/null_pointer.c:238	Not processed
 01.w_Defects/null_pointer.c:334	Not processed
 01.w_Defects/st_cross_thread_access.c:335	Not processed
 01.w_Defects/st_underrun.c:80	Not processed
 01.w_Defects/st_underrun.c:115	Not processed
 01.w_Defects/st_underrun.c:204	Not processed
 01.w_Defects/uninit_memory_access.c:54	Not processed
 01.w_Defects/uninit_pointer.c:187	Not processed
 01.w_Defects/uninit_pointer.c:385	Not processed
 01.w_Defects/uninit_pointer.c:406	Not processed
 01.w_Defects/uninit_var.c:142	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:213	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:214	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:382	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:439	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:478	Not processed
 02.wo_Defects/free_null_pointer.c:26	Not processed
 02.wo_Defects/free_null_pointer.c:119	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Insecure string API

C/C++

 02.wo_Defects/free_null_pointer.c:156	Not processed
 02.wo_Defects/free_null_pointer.c:249	Not processed
 02.wo_Defects/free_null_pointer.c:284	Not processed
 02.wo_Defects/free_null_pointer.c:325	Not processed
 02.wo_Defects/invalid_memory_access.c:105	Not processed
 02.wo_Defects/invalid_memory_access.c:213	Not processed
 02.wo_Defects/invalid_memory_access.c:567	Not processed
 02.wo_Defects/invalid_memory_access.c:576	Not processed
 02.wo_Defects/invalid_memory_access.c:622	Not processed
 02.wo_Defects/invalid_memory_access.c:634	Not processed
 02.wo_Defects/memory_allocation_failure.c:594	Not processed
 02.wo_Defects/memory_leak.c:76	Not processed
 02.wo_Defects/memory_leak.c:100	Not processed
 02.wo_Defects/memory_leak.c:277	Not processed
 02.wo_Defects/memory_leak.c:408	Not processed
 02.wo_Defects/memory_leak.c:431	Not processed
 02.wo_Defects/memory_leak.c:526	Not processed
 02.wo_Defects/null_pointer.c:259	Not processed
 02.wo_Defects/null_pointer.c:354	Not processed
 02.wo_Defects/st_underrun.c:81	Not processed
 02.wo_Defects/st_underrun.c:116	Not processed
 02.wo_Defects/st_underrun.c:213	Not processed
 02.wo_Defects/uninit_memory_access.c:55	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Insecure string API

C/C++

	02.wo_Defects/uninit_pointer.c:197	Not processed
	02.wo_Defects/uninit_pointer.c:400	Not processed
	02.wo_Defects/uninit_pointer.c:423	Not processed
	02.wo_Defects/uninit_var.c:152	Not processed
	02.wo_Defects/wrong_arguments_func_pointer.c:208	Not processed
	02.wo_Defects/wrong_arguments_func_pointer.c:209	Not processed
	02.wo_Defects/wrong_arguments_func_pointer.c:381	Not processed
	02.wo_Defects/wrong_arguments_func_pointer.c:435	Not processed
	02.wo_Defects/wrong_arguments_func_pointer.c:474	Not processed

Invalid memory deallocation function usage

C/C++

17

	01.w_Defects/free_nondynamically_allocated_memory.c:22	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:36	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:48	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:62	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:86	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:103	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:115	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:128	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:141	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:155	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:170	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:187	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:209	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Invalid memory deallocation function usage

C/C++

01.w_Defects/free_nondynamically_allocated_memory.c:229	Not processed
01.w_Defects/free_nondynamically_allocated_memory.c:239	Not processed
01.w_Defects/free_nondynamically_allocated_memory.c:262	Not processed
01.w_Defects/uninit_memory_access.c:420	Not processed

Loss of precision

C/C++

34

01.w_Defects/bit_shift.c:35	Not processed
01.w_Defects/bit_shift.c:59	Not processed
01.w_Defects/data_lost.c:22	Not processed
01.w_Defects/data_lost.c:34	Not processed
01.w_Defects/data_lost.c:46	Not processed
01.w_Defects/data_lost.c:94	Not processed
01.w_Defects/data_lost.c:106	Not processed
01.w_Defects/data_lost.c:118	Not processed
01.w_Defects/data_lost.c:209	Not processed
01.w_Defects/data_lost.c:240	Not processed
01.w_Defects/data_lost.c:256	Not processed
01.w_Defects/data_overflow.c:62	Not processed
01.w_Defects/data_overflow.c:110	Not processed
01.w_Defects/data_underflow.c:34	Not processed
01.w_Defects/sign_conv.c:23	Not processed
01.w_Defects/sign_conv.c:35	Not processed
01.w_Defects/sign_conv.c:47	Not processed
01.w_Defects/sign_conv.c:60	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Loss of precision

C/C++

 01.w_Defects/sign_conv.c:71	Not processed
 01.w_Defects/sign_conv.c:83	Not processed
 01.w_Defects/sign_conv.c:95	Not processed
 01.w_Defects/sign_conv.c:96	Not processed
 01.w_Defects/sign_conv.c:107	Not processed
 01.w_Defects/sign_conv.c:108	Not processed
 01.w_Defects/sign_conv.c:151	Not processed
 01.w_Defects/sign_conv.c:180	Not processed
 01.w_Defects/sign_conv.c:192	Not processed
 01.w_Defects/sign_conv.c:217	Not processed
 01.w_Defects/sign_conv.c:246	Not processed
 01.w_Defects/sign_conv.c:247	Not processed
 01.w_Defects/sign_conv.c:262	Not processed
 01.w_Defects/sign_conv.c:263	Not processed
 02.wo_Defects/data_overflow.c:99	Not processed
 02.wo_Defects/data_overflow.c:111	Not processed

Malloc overflow

C/C++

18

 01.w_Defects/free_null_pointer.c:102	Not processed
 01.w_Defects/free_null_pointer.c:231	Not processed
 01.w_Defects/invalid_memory_access.c:280	Not processed
 01.w_Defects/memory_allocation_failure.c:258	Not processed
 01.w_Defects/memory_leak.c:64	Not processed
 01.w_Defects/memory_leak.c:392	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Malloc overflow

C/C++

01.w_Defects/memory_leak.c:417	Not processed
01.w_Defects/null_pointer.c:231	Not processed
01.w_Defects/wrong_arguments_func_pointer.c:371	Not processed
02.wo_Defects/free_null_pointer.c:112	Not processed
02.wo_Defects/free_null_pointer.c:239	Not processed
02.wo_Defects/invalid_memory_access.c:289	Not processed
02.wo_Defects/memory_allocation_failure.c:267	Not processed
02.wo_Defects/memory_leak.c:67	Not processed
02.wo_Defects/memory_leak.c:398	Not processed
02.wo_Defects/memory_leak.c:424	Not processed
02.wo_Defects/null_pointer.c:252	Not processed
02.wo_Defects/wrong_arguments_func_pointer.c:370	Not processed

Memory leak

C/C++

37

01.w_Defects/double_free.c:88	Not processed
01.w_Defects/func_pointer.c:117	Not processed
01.w_Defects/func_pointer.c:411	Not processed
01.w_Defects/func_pointer.c:608	Not processed
01.w_Defects/memory_allocation_failure.c:283	Not processed
01.w_Defects/memory_leak.c:74	Not processed
01.w_Defects/memory_leak.c:120	Not processed
01.w_Defects/memory_leak.c:151	Not processed
01.w_Defects/memory_leak.c:219	Not processed
01.w_Defects/memory_leak.c:236	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Memory leak

C/C++

	01.w_Defects/memory_leak.c:248	Not processed
	01.w_Defects/memory_leak.c:276	Not processed
	01.w_Defects/memory_leak.c:311	Not processed
	01.w_Defects/memory_leak.c:359	Not processed
	01.w_Defects/memory_leak.c:384	Not processed
	01.w_Defects/memory_leak.c:404	Not processed
	01.w_Defects/uninit_memory_access.c:201	Not processed
	01.w_Defects/uninit_pointer.c:125	Not processed
	01.w_Defects/uninit_pointer.c:125	Not processed
	01.w_Defects/uninit_pointer.c:125	Not processed
	01.w_Defects/uninit_pointer.c:125	Not processed
	01.w_Defects/uninit_pointer.c:236	Not processed
	01.w_Defects/wrong_arguments_func_pointer.c:601	Not processed
	02.wo_Defects/free_null_pointer.c:491	Not processed
	02.wo_Defects/func_pointer.c:635	Not processed
	02.wo_Defects/memory_allocation_failure.c:293	Not processed
	02.wo_Defects/uninit_memory_access.c:131	Not processed
	02.wo_Defects/uninit_pointer.c:132	Not processed
	02.wo_Defects/uninit_pointer.c:132	Not processed
	02.wo_Defects/uninit_pointer.c:132	Not processed
	02.wo_Defects/uninit_pointer.c:132	Not processed
	02.wo_Defects/uninit_pointer.c:248	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Memory leak

C/C++

02.wo_Defects/wrong_arguments_func_pointer.c:218	Not processed
02.wo_Defects/wrong_arguments_func_pointer.c:220	Not processed
02.wo_Defects/wrong_arguments_func_pointer.c:357	Not processed
02.wo_Defects/wrong_arguments_func_pointer.c:596	Not processed

Mutex double lock

C/C++

9

01.w_Defects/double_lock.c:42	Not processed
01.w_Defects/double_lock.c:93	Not processed
01.w_Defects/double_lock.c:140	Not processed
01.w_Defects/livelock.c:32	Not processed
01.w_Defects/livelock.c:49	Not processed
01.w_Defects/lock_never_unlock.c:93	Not processed
01.w_Defects/lock_never_unlock.c:147	Not processed
01.w_Defects/lock_never_unlock.c:398	Not processed
01.w_Defects/lock_never_unlock.c:551	Not processed

Mutex double unlock

C/C++

12

01.w_Defects/double_release.c:35	Not processed
01.w_Defects/double_release.c:133	Not processed
01.w_Defects/double_release.c:178	Not processed
01.w_Defects/double_release.c:226	Not processed
01.w_Defects/double_release.c:283	Not processed
01.w_Defects/unlock_without_lock.c:110	Not processed
01.w_Defects/unlock_without_lock.c:202	Not processed
01.w_Defects/unlock_without_lock.c:281	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Mutex double unlock

C/C++

01.w_Defects/unlock_without_lock.c:374	Not processed
01.w_Defects/unlock_without_lock.c:544	Not processed
02.wo_Defects/livelock.c:33	Not processed
02.wo_Defects/livelock.c:52	Not processed

Null pointer dereference

C/C++

36

01.w_Defects/buffer_underrun_dynamic.c:155	Not processed
01.w_Defects/free_nondynamically_allocated_memory.c:82	Not processed
01.w_Defects/free_null_pointer.c:275	Not processed
01.w_Defects/free_null_pointer.c:452	Not processed
01.w_Defects/free_null_pointer.c:483	Not processed
01.w_Defects/func_pointer.c:177	Not processed
01.w_Defects/func_pointer.c:352	Not processed
01.w_Defects/invalid_memory_access.c:45	Not processed
01.w_Defects/invalid_memory_access.c:181	Not processed
01.w_Defects/invalid_memory_access.c:265	Not processed
01.w_Defects/invalid_memory_access.c:320	Not processed
01.w_Defects/invalid_memory_access.c:371	Not processed
01.w_Defects/invalid_memory_access.c:432	Not processed
01.w_Defects/memory_allocation_failure.c:82	Not processed
01.w_Defects/memory_leak.c:96	Not processed
01.w_Defects/null_pointer.c:23	Not processed
01.w_Defects/null_pointer.c:34	Not processed
01.w_Defects/null_pointer.c:47	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Null pointer dereference

C/C++

 01.w_Defects/null_pointer.c:63	Not processed
 01.w_Defects/null_pointer.c:94	Not processed
 01.w_Defects/null_pointer.c:117	Not processed
 01.w_Defects/null_pointer.c:142	Not processed
 01.w_Defects/null_pointer.c:159	Not processed
 01.w_Defects/null_pointer.c:173	Not processed
 01.w_Defects/null_pointer.c:180	Not processed
 01.w_Defects/null_pointer.c:334	Not processed
 01.w_Defects/uninit_memory_access.c:74	Not processed
 01.w_Defects/uninit_pointer.c:131	Not processed
 02.wo_Defects/buffer_underrun_dynamic.c:723	Not processed
 02.wo_Defects/free_nondynamically_allocated_memory.c:82	Not processed
 02.wo_Defects/invalid_memory_access.c:70	Not processed
 02.wo_Defects/invalid_memory_access.c:183	Not processed
 02.wo_Defects/invalid_memory_access.c:275	Not processed
 02.wo_Defects/memory_allocation_failure.c:84	Not processed
 02.wo_Defects/memory_leak.c:100	Not processed
 02.wo_Defects/uninit_memory_access.c:76	Not processed

Reference to array element out of bounds

C/C++

152

 01.w_Defects/buffer_overrun_dynamic.c:27	Not processed
 01.w_Defects/buffer_overrun_dynamic.c:42	Not processed
 01.w_Defects/buffer_overrun_dynamic.c:62	Not processed
 01.w_Defects/buffer_overrun_dynamic.c:77	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Reference to array element out of bounds

C/C++

	01.w_Defects/buffer_overrun_dynamic.c:94	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:112	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:130	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:152	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:174	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:233	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:248	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:263	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:278	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:298	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:312	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:333	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:350	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:369	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:387	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:403	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:422	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:435	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:462	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:480	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:496	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:514	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:532	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Reference to array element out of bounds

C/C++

	01.w_Defects/buffer_overrun_dynamic.c:559	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:580	Not processed
	01.w_Defects/buffer_overrun_dynamic.c:607	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:29	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:45	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:65	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:80	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:97	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:115	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:133	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:155	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:178	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:237	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:268	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:283	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:303	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:317	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:338	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:355	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:374	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:392	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:408	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:427	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Reference to array element out of bounds

C/C++

	01.w_Defects/buffer_underrun_dynamic.c:439	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:466	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:484	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:500	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:519	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:532	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:559	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:580	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:606	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:621	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:624	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:648	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:674	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:701	Not processed
	01.w_Defects/buffer_underrun_dynamic.c:721	Not processed
	01.w_Defects/func_pointer.c:395	Not processed
	01.w_Defects/invalid_memory_access.c:320	Not processed
	01.w_Defects/overrun_st.c:21	Not processed
	01.w_Defects/overrun_st.c:32	Not processed
	01.w_Defects/overrun_st.c:44	Not processed
	01.w_Defects/overrun_st.c:55	Not processed
	01.w_Defects/overrun_st.c:66	Not processed
	01.w_Defects/overrun_st.c:77	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Reference to array element out of bounds

C/C++

 01.w_Defects/overrun_st.c:88	Not processed
 01.w_Defects/overrun_st.c:99	Not processed
 01.w_Defects/overrun_st.c:110	Not processed
 01.w_Defects/overrun_st.c:126	Not processed
 01.w_Defects/overrun_st.c:158	Not processed
 01.w_Defects/overrun_st.c:169	Not processed
 01.w_Defects/overrun_st.c:194	Not processed
 01.w_Defects/overrun_st.c:206	Not processed
 01.w_Defects/overrun_st.c:222	Not processed
 01.w_Defects/overrun_st.c:233	Not processed
 01.w_Defects/overrun_st.c:250	Not processed
 01.w_Defects/overrun_st.c:264	Not processed
 01.w_Defects/overrun_st.c:280	Not processed
 01.w_Defects/overrun_st.c:293	Not processed
 01.w_Defects/overrun_st.c:306	Not processed
 01.w_Defects/overrun_st.c:320	Not processed
 01.w_Defects/overrun_st.c:333	Not processed
 01.w_Defects/overrun_st.c:346	Not processed
 01.w_Defects/overrun_st.c:359	Not processed
 01.w_Defects/overrun_st.c:372	Not processed
 01.w_Defects/overrun_st.c:387	Not processed
 01.w_Defects/overrun_st.c:402	Not processed
 01.w_Defects/overrun_st.c:415	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Reference to array element out of bounds

C/C++

	01.w_Defects/overrun_st.c:428	Not processed
	01.w_Defects/overrun_st.c:457	Not processed
	01.w_Defects/overrun_st.c:471	Not processed
	01.w_Defects/overrun_st.c:489	Not processed
	01.w_Defects/overrun_st.c:502	Not processed
	01.w_Defects/overrun_st.c:522	Not processed
	01.w_Defects/overrun_st.c:538	Not processed
	01.w_Defects/overrun_st.c:556	Not processed
	01.w_Defects/overrun_st.c:570	Not processed
	01.w_Defects/overrun_st.c:588	Not processed
	01.w_Defects/overrun_st.c:613	Not processed
	01.w_Defects/overrun_st.c:630	Not processed
	01.w_Defects/overrun_st.c:642	Not processed
	01.w_Defects/overrun_st.c:658	Not processed
	01.w_Defects/overrun_st.c:674	Not processed
	01.w_Defects/overrun_st.c:689	Not processed
	01.w_Defects/overrun_st.c:706	Not processed
	01.w_Defects/overrun_st.c:724	Not processed
	01.w_Defects/overrun_st.c:739	Not processed
	01.w_Defects/overrun_st.c:749	Not processed
	01.w_Defects/overrun_st.c:761	Not processed
	01.w_Defects/overrun_st.c:773	Not processed
	01.w_Defects/st_underrun.c:25	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Reference to array element out of bounds

C/C++

	01.w_Defects/st_underrun.c:51	Not processed
	01.w_Defects/st_underrun.c:88	Not processed
	01.w_Defects/st_underrun.c:92	Not processed
	01.w_Defects/st_underrun.c:125	Not processed
	01.w_Defects/st_underrun.c:126	Not processed
	01.w_Defects/st_underrun.c:126	Not processed
	01.w_Defects/st_underrun.c:130	Not processed
	01.w_Defects/st_underrun.c:150	Not processed
	01.w_Defects/st_underrun.c:160	Not processed
	01.w_Defects/st_underrun.c:193	Not processed
	01.w_Defects/st_underrun.c:195	Not processed
	01.w_Defects/st_underrun.c:227	Not processed
	01.w_Defects/st_underrun.c:229	Not processed
	01.w_Defects/underrun_st.c:21	Not processed
	01.w_Defects/underrun_st.c:31	Not processed
	01.w_Defects/underrun_st.c:42	Not processed
	01.w_Defects/underrun_st.c:55	Not processed
	01.w_Defects/underrun_st.c:67	Not processed
	01.w_Defects/underrun_st.c:80	Not processed
	01.w_Defects/underrun_st.c:93	Not processed
	01.w_Defects/underrun_st.c:109	Not processed
	01.w_Defects/underrun_st.c:124	Not processed
	01.w_Defects/underrun_st.c:140	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Reference to array element out of bounds

C/C++

01.w_Defects/underrun_st.c:155	Not processed
01.w_Defects/underrun_st.c:172	Not processed
01.w_Defects/underrun_st.c:190	Not processed
01.w_Defects/wrong_arguments_func_pointer.c:271	Not processed
01.w_Defects/wrong_arguments_func_pointer.c:324	Not processed
02.wo_Defects/func_pointer.c:414	Not processed
02.wo_Defects/st_underrun.c:52	Not processed
02.wo_Defects/st_underrun.c:236	Not processed
02.wo_Defects/st_underrun.c:238	Not processed
02.wo_Defects/wrong_arguments_func_pointer.c:409	Not processed

Type cast error

C/C++

20

01.w_Defects/littlemem_st.c:35	Not processed
01.w_Defects/littlemem_st.c:54	Not processed
01.w_Defects/littlemem_st.c:91	Not processed
01.w_Defects/littlemem_st.c:111	Not processed
01.w_Defects/littlemem_st.c:137	Not processed
01.w_Defects/littlemem_st.c:171	Not processed
01.w_Defects/littlemem_st.c:214	Not processed
01.w_Defects/littlemem_st.c:258	Not processed
01.w_Defects/littlemem_st.c:300	Not processed
01.w_Defects/littlemem_st.c:332	Not processed
02.wo_Defects/littlemem_st.c:36	Not processed
02.wo_Defects/littlemem_st.c:55	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Type cast error

C/C++

02.wo_Defects/littlemem_st.c:92	Not processed
02.wo_Defects/littlemem_st.c:112	Not processed
02.wo_Defects/littlemem_st.c:138	Not processed
02.wo_Defects/littlemem_st.c:172	Not processed
02.wo_Defects/littlemem_st.c:215	Not processed
02.wo_Defects/littlemem_st.c:259	Not processed
02.wo_Defects/littlemem_st.c:301	Not processed
02.wo_Defects/littlemem_st.c:333	Not processed

Undefined result

C/C++

27

01.w_Defects/bit_shift.c:22	Not processed
01.w_Defects/bit_shift.c:46	Not processed
01.w_Defects/bit_shift.c:70	Not processed
01.w_Defects/bit_shift.c:82	Not processed
01.w_Defects/bit_shift.c:94	Not processed
01.w_Defects/bit_shift.c:107	Not processed
01.w_Defects/bit_shift.c:134	Not processed
01.w_Defects/bit_shift.c:147	Not processed
01.w_Defects/bit_shift.c:164	Not processed
01.w_Defects/bit_shift.c:176	Not processed
01.w_Defects/bit_shift.c:194	Not processed
01.w_Defects/bit_shift.c:209	Not processed
01.w_Defects/bit_shift.c:226	Not processed
01.w_Defects/bit_shift.c:237	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Undefined result

C/C++

 01.w_Defects/zero_division.c:23	Not processed
 01.w_Defects/zero_division.c:34	Not processed
 01.w_Defects/zero_division.c:47	Not processed
 01.w_Defects/zero_division.c:78	Not processed
 01.w_Defects/zero_division.c:118	Not processed
 01.w_Defects/zero_division.c:141	Not processed
 01.w_Defects/zero_division.c:166	Not processed
 01.w_Defects/zero_division.c:178	Not processed
 01.w_Defects/zero_division.c:195	Not processed
 01.w_Defects/zero_division.c:206	Not processed
 01.w_Defects/zero_division.c:225	Not processed
 01.w_Defects/zero_division.c:252	Not processed
 03.w_Defects_Cpp/improper_error_handling.cpp:22	Not processed

Uninitialized variable

C/C++

55

 01.w_Defects/buffer_overrun_dynamic.c:333	Not processed
 01.w_Defects/buffer_underrun_dynamic.c:178	Not processed
 01.w_Defects/buffer_underrun_dynamic.c:648	Not processed
 01.w_Defects/buffer_underrun_dynamic.c:674	Not processed
 01.w_Defects/buffer_underrun_dynamic.c:722	Not processed
 01.w_Defects/buffer_underrun_dynamic.c:724	Not processed
 01.w_Defects/func_pointer.c:262	Not processed
 01.w_Defects/func_pointer.c:262	Not processed
 01.w_Defects/memory_leak.c:276	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Uninitialized variable

C/C++

 01.w_Defects/overrun_st.c:44	Not processed
 01.w_Defects/overrun_st.c:320	Not processed
 01.w_Defects/ow_memcpy.c:41	Not processed
 01.w_Defects/st_underrun.c:227	Not processed
 01.w_Defects/st_underrun.c:229	Not processed
 01.w_Defects/underrun_st.c:21	Not processed
 01.w_Defects/underrun_st.c:55	Not processed
 01.w_Defects/uninit_memory_access.c:98	Not processed
 01.w_Defects/uninit_memory_access.c:200	Not processed
 01.w_Defects/uninit_memory_access.c:249	Not processed
 01.w_Defects/uninit_memory_access.c:249	Not processed
 01.w_Defects/uninit_memory_access.c:298	Not processed
 01.w_Defects/uninit_pointer.c:30	Not processed
 01.w_Defects/uninit_pointer.c:41	Not processed
 01.w_Defects/uninit_pointer.c:55	Not processed
 01.w_Defects/uninit_pointer.c:90	Not processed
 01.w_Defects/uninit_pointer.c:104	Not processed
 01.w_Defects/uninit_pointer.c:131	Not processed
 01.w_Defects/uninit_pointer.c:152	Not processed
 01.w_Defects/uninit_pointer.c:200	Not processed
 01.w_Defects/uninit_pointer.c:200	Not processed
 01.w_Defects/uninit_pointer.c:231	Not processed
 01.w_Defects/uninit_pointer.c:231	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Uninitialized variable

C/C++

 01.w_Defects/uninit_pointer.c:254	Not processed
 01.w_Defects/uninit_pointer.c:254	Not processed
 01.w_Defects/uninit_var.c:23	Not processed
 01.w_Defects/uninit_var.c:34	Not processed
 01.w_Defects/uninit_var.c:45	Not processed
 01.w_Defects/uninit_var.c:63	Not processed
 01.w_Defects/uninit_var.c:75	Not processed
 01.w_Defects/uninit_var.c:92	Not processed
 01.w_Defects/uninit_var.c:111	Not processed
 01.w_Defects/uninit_var.c:161	Not processed
 01.w_Defects/uninit_var.c:177	Not processed
 01.w_Defects/uninit_var.c:178	Not processed
 01.w_Defects/uninit_var.c:243	Not processed
 01.w_Defects/uninit_var.c:296	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:337	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:337	Not processed
 01.w_Defects/zero_division.c:34	Not processed
 01.w_Defects/zero_division.c:35	Not processed
 02.wo_Defects/ow_memcpy.c:42	Not processed
 02.wo_Defects/st_underrun.c:236	Not processed
 02.wo_Defects/st_underrun.c:238	Not processed
 02.wo_Defects/st_underrun.c:242	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:409	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Unsafe function: rand

C/C++

97

 01.w_Defects/bit_shift.c:120	Not processed
 01.w_Defects/buffer_underrun_dynamic.c:249	Not processed
 01.w_Defects/conflicting_cond.c:23	Not processed
 01.w_Defects/conflicting_cond.c:42	Not processed
 01.w_Defects/conflicting_cond.c:61	Not processed
 01.w_Defects/conflicting_cond.c:80	Not processed
 01.w_Defects/conflicting_cond.c:102	Not processed
 01.w_Defects/conflicting_cond.c:136	Not processed
 01.w_Defects/conflicting_cond.c:156	Not processed
 01.w_Defects/conflicting_cond.c:176	Not processed
 01.w_Defects/conflicting_cond.c:197	Not processed
 01.w_Defects/data_lost.c:157	Not processed
 01.w_Defects/data_overflow.c:204	Not processed
 01.w_Defects/double_free.c:81	Not processed
 01.w_Defects/double_free.c:86	Not processed
 01.w_Defects/double_release.c:54	Not processed
 01.w_Defects/double_release.c:105	Not processed
 01.w_Defects/double_release.c:151	Not processed
 01.w_Defects/double_release.c:196	Not processed
 01.w_Defects/double_release.c:245	Not processed
 01.w_Defects/double_release.c:281	Not processed
 01.w_Defects/double_release.c:292	Not processed
 01.w_Defects/free_null_pointer.c:392	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Unsafe function: rand

C/C++

	01.w_Defects/func_pointer.c:330	Not processed
	01.w_Defects/insign_code.c:22	Not processed
	01.w_Defects/memory_allocation_failure.c:150	Not processed
	01.w_Defects/memory_leak.c:197	Not processed
	01.w_Defects/not_return.c:29	Not processed
	01.w_Defects/not_return.c:55	Not processed
	01.w_Defects/not_return.c:81	Not processed
	01.w_Defects/not_return.c:104	Not processed
	01.w_Defects/null_pointer.c:104	Not processed
	01.w_Defects/overrun_st.c:181	Not processed
	01.w_Defects/overrun_st.c:442	Not processed
	01.w_Defects/redundant_cond.c:25	Not processed
	01.w_Defects/redundant_cond.c:44	Not processed
	01.w_Defects/redundant_cond.c:63	Not processed
	01.w_Defects/redundant_cond.c:82	Not processed
	01.w_Defects/redundant_cond.c:101	Not processed
	01.w_Defects/redundant_cond.c:120	Not processed
	01.w_Defects/redundant_cond.c:142	Not processed
	01.w_Defects/redundant_cond.c:176	Not processed
	01.w_Defects/redundant_cond.c:196	Not processed
	01.w_Defects/redundant_cond.c:216	Not processed
	01.w_Defects/redundant_cond.c:236	Not processed
	01.w_Defects/redundant_cond.c:256	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Unsafe function: rand

C/C++

	01.w_Defects/redundant_cond.c:276	Not processed
	01.w_Defects/sign_conv.c:165	Not processed
	01.w_Defects/sleep_lock.c:70	Not processed
	01.w_Defects/sleep_lock.c:148	Not processed
	01.w_Defects/sleep_lock.c:202	Not processed
	01.w_Defects/uninit_memory_access.c:385	Not processed
	01.w_Defects/zero_division.c:153	Not processed
	02.wo_Defects/bit_shift.c:120	Not processed
	02.wo_Defects/conflicting_cond.c:24	Not processed
	02.wo_Defects/conflicting_cond.c:43	Not processed
	02.wo_Defects/conflicting_cond.c:62	Not processed
	02.wo_Defects/conflicting_cond.c:81	Not processed
	02.wo_Defects/conflicting_cond.c:104	Not processed
	02.wo_Defects/conflicting_cond.c:138	Not processed
	02.wo_Defects/conflicting_cond.c:158	Not processed
	02.wo_Defects/conflicting_cond.c:178	Not processed
	02.wo_Defects/conflicting_cond.c:199	Not processed
	02.wo_Defects/data_lost.c:160	Not processed
	02.wo_Defects/data_overflow.c:205	Not processed
	02.wo_Defects/double_release.c:53	Not processed
	02.wo_Defects/double_release.c:106	Not processed
	02.wo_Defects/double_release.c:154	Not processed
	02.wo_Defects/double_release.c:201	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Unsafe function: rand

C/C++

 02.wo_Defects/double_release.c:250	Not processed
 02.wo_Defects/double_release.c:292	Not processed
 02.wo_Defects/free_null_pointer.c:364	Not processed
 02.wo_Defects/func_pointer.c:341	Not processed
 02.wo_Defects/insign_code.c:23	Not processed
 02.wo_Defects/memory_leak.c:200	Not processed
 02.wo_Defects/not_return.c:34	Not processed
 02.wo_Defects/not_return.c:61	Not processed
 02.wo_Defects/not_return.c:87	Not processed
 02.wo_Defects/not_return.c:111	Not processed
 02.wo_Defects/overrun_st.c:183	Not processed
 02.wo_Defects/overrun_st.c:443	Not processed
 02.wo_Defects/redundant_cond.c:26	Not processed
 02.wo_Defects/redundant_cond.c:45	Not processed
 02.wo_Defects/redundant_cond.c:64	Not processed
 02.wo_Defects/redundant_cond.c:83	Not processed
 02.wo_Defects/redundant_cond.c:102	Not processed
 02.wo_Defects/redundant_cond.c:121	Not processed
 02.wo_Defects/redundant_cond.c:141	Not processed
 02.wo_Defects/redundant_cond.c:175	Not processed
 02.wo_Defects/redundant_cond.c:195	Not processed
 02.wo_Defects/redundant_cond.c:215	Not processed
 02.wo_Defects/redundant_cond.c:235	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Unsafe function: rand

C/C++

02.wo_Defects/redundant_cond.c:255	Not processed
02.wo_Defects/redundant_cond.c:275	Not processed
02.wo_Defects/sign_conv.c:165	Not processed
02.wo_Defects/uninit_memory_access.c:401	Not processed
02.wo_Defects/zero_division.c:151	Not processed

Use after free

C/C++

15

01.w_Defects/invalid_memory_access.c:45	Not processed
01.w_Defects/invalid_memory_access.c:84	Not processed
01.w_Defects/invalid_memory_access.c:133	Not processed
01.w_Defects/invalid_memory_access.c:188	Not processed
01.w_Defects/invalid_memory_access.c:210	Not processed
01.w_Defects/invalid_memory_access.c:224	Not processed
01.w_Defects/invalid_memory_access.c:270	Not processed
01.w_Defects/invalid_memory_access.c:294	Not processed
01.w_Defects/invalid_memory_access.c:320	Not processed
01.w_Defects/invalid_memory_access.c:371	Not processed
01.w_Defects/invalid_memory_access.c:432	Not processed
01.w_Defects/invalid_memory_access.c:516	Not processed
01.w_Defects/invalid_memory_access.c:568	Not processed
01.w_Defects/invalid_memory_access.c:622	Not processed
02.wo_Defects/buffer_underrun_dynamic.c:723	Not processed

Weak random number generator

C/C++

97

01.w_Defects/bit_shift.c:120	Not processed
------------------------------	---------------

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Weak random number generator

C/C++

01.w_Defects/buffer_underrun_dynamic.c:249	Not processed
01.w_Defects/conflicting_cond.c:23	Not processed
01.w_Defects/conflicting_cond.c:42	Not processed
01.w_Defects/conflicting_cond.c:61	Not processed
01.w_Defects/conflicting_cond.c:80	Not processed
01.w_Defects/conflicting_cond.c:102	Not processed
01.w_Defects/conflicting_cond.c:136	Not processed
01.w_Defects/conflicting_cond.c:156	Not processed
01.w_Defects/conflicting_cond.c:176	Not processed
01.w_Defects/conflicting_cond.c:197	Not processed
01.w_Defects/data_lost.c:157	Not processed
01.w_Defects/data_overflow.c:204	Not processed
01.w_Defects/double_free.c:81	Not processed
01.w_Defects/double_free.c:86	Not processed
01.w_Defects/double_release.c:54	Not processed
01.w_Defects/double_release.c:105	Not processed
01.w_Defects/double_release.c:151	Not processed
01.w_Defects/double_release.c:196	Not processed
01.w_Defects/double_release.c:245	Not processed
01.w_Defects/double_release.c:281	Not processed
01.w_Defects/double_release.c:292	Not processed
01.w_Defects/free_null_pointer.c:392	Not processed
01.w_Defects/func_pointer.c:330	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Weak random number generator

C/C++

	01.w_Defects/insign_code.c:22	Not processed
	01.w_Defects/memory_allocation_failure.c:150	Not processed
	01.w_Defects/memory_leak.c:197	Not processed
	01.w_Defects/not_return.c:29	Not processed
	01.w_Defects/not_return.c:55	Not processed
	01.w_Defects/not_return.c:81	Not processed
	01.w_Defects/not_return.c:104	Not processed
	01.w_Defects/null_pointer.c:104	Not processed
	01.w_Defects/overrun_st.c:181	Not processed
	01.w_Defects/overrun_st.c:442	Not processed
	01.w_Defects/redundant_cond.c:25	Not processed
	01.w_Defects/redundant_cond.c:44	Not processed
	01.w_Defects/redundant_cond.c:63	Not processed
	01.w_Defects/redundant_cond.c:82	Not processed
	01.w_Defects/redundant_cond.c:101	Not processed
	01.w_Defects/redundant_cond.c:120	Not processed
	01.w_Defects/redundant_cond.c:142	Not processed
	01.w_Defects/redundant_cond.c:176	Not processed
	01.w_Defects/redundant_cond.c:196	Not processed
	01.w_Defects/redundant_cond.c:216	Not processed
	01.w_Defects/redundant_cond.c:236	Not processed
	01.w_Defects/redundant_cond.c:256	Not processed
	01.w_Defects/redundant_cond.c:276	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Weak random number generator

C/C++

01.w_Defects/sign_conv.c:165	Not processed
01.w_Defects/sleep_lock.c:70	Not processed
01.w_Defects/sleep_lock.c:148	Not processed
01.w_Defects/sleep_lock.c:202	Not processed
01.w_Defects/uninit_memory_access.c:385	Not processed
01.w_Defects/zero_division.c:153	Not processed
02.wo_Defects/bit_shift.c:120	Not processed
02.wo_Defects/conflicting_cond.c:24	Not processed
02.wo_Defects/conflicting_cond.c:43	Not processed
02.wo_Defects/conflicting_cond.c:62	Not processed
02.wo_Defects/conflicting_cond.c:81	Not processed
02.wo_Defects/conflicting_cond.c:104	Not processed
02.wo_Defects/conflicting_cond.c:138	Not processed
02.wo_Defects/conflicting_cond.c:158	Not processed
02.wo_Defects/conflicting_cond.c:178	Not processed
02.wo_Defects/conflicting_cond.c:199	Not processed
02.wo_Defects/data_lost.c:160	Not processed
02.wo_Defects/data_overflow.c:205	Not processed
02.wo_Defects/double_release.c:53	Not processed
02.wo_Defects/double_release.c:106	Not processed
02.wo_Defects/double_release.c:154	Not processed
02.wo_Defects/double_release.c:201	Not processed
02.wo_Defects/double_release.c:250	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Weak random number generator

C/C++

	02.wo_Defects/double_release.c:292	Not processed
	02.wo_Defects/free_null_pointer.c:364	Not processed
	02.wo_Defects/func_pointer.c:341	Not processed
	02.wo_Defects/insign_code.c:23	Not processed
	02.wo_Defects/memory_leak.c:200	Not processed
	02.wo_Defects/not_return.c:34	Not processed
	02.wo_Defects/not_return.c:61	Not processed
	02.wo_Defects/not_return.c:87	Not processed
	02.wo_Defects/not_return.c:111	Not processed
	02.wo_Defects/overrun_st.c:183	Not processed
	02.wo_Defects/overrun_st.c:443	Not processed
	02.wo_Defects/redundant_cond.c:26	Not processed
	02.wo_Defects/redundant_cond.c:45	Not processed
	02.wo_Defects/redundant_cond.c:64	Not processed
	02.wo_Defects/redundant_cond.c:83	Not processed
	02.wo_Defects/redundant_cond.c:102	Not processed
	02.wo_Defects/redundant_cond.c:121	Not processed
	02.wo_Defects/redundant_cond.c:141	Not processed
	02.wo_Defects/redundant_cond.c:175	Not processed
	02.wo_Defects/redundant_cond.c:195	Not processed
	02.wo_Defects/redundant_cond.c:215	Not processed
	02.wo_Defects/redundant_cond.c:235	Not processed
	02.wo_Defects/redundant_cond.c:255	Not processed

* Rejected vulnerabilities are not taken into account

Medium-level vulnerabilities

Weak random number generator

C/C++

	02.wo_Defects/redundant_cond.c:275	Not processed
	02.wo_Defects/sign_conv.c:165	Not processed
	02.wo_Defects/uninit_memory_access.c:401	Not processed
	02.wo_Defects/zero_division.c:151	Not processed

Low-level vulnerabilities

232*

Block critical section

C/C++

4

	01.w_Defects/sleep_lock.c:50	Not processed
	01.w_Defects/sleep_lock.c:123	Not processed
	01.w_Defects/sleep_lock.c:124	Not processed
	01.w_Defects/sleep_lock.c:173	Not processed

Dead store

C/C++

159

	01.w_Defects/buffer_underrun_dynamic.c:701	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:98	Not processed
	01.w_Defects/free_nondynamically_allocated_memory.c:99	Not processed
	01.w_Defects/free_null_pointer.c:419	Not processed
	01.w_Defects/free_null_pointer.c:420	Not processed
	01.w_Defects/free_null_pointer.c:452	Not processed
	01.w_Defects/function_return_value_unchecked.c:393	Not processed
	01.w_Defects/function_return_value_unchecked.c:398	Not processed
	01.w_Defects/func_pointer.c:34	Not processed
	01.w_Defects/func_pointer.c:42	Not processed
	01.w_Defects/func_pointer.c:81	Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Dead store

C/C++

 01.w_Defects/func_pointer.c:123	Not processed
 01.w_Defects/func_pointer.c:256	Not processed
 01.w_Defects/func_pointer.c:265	Not processed
 01.w_Defects/func_pointer.c:285	Not processed
 01.w_Defects/func_pointer.c:330	Not processed
 01.w_Defects/func_pointer.c:353	Not processed
 01.w_Defects/func_pointer.c:365	Not processed
 01.w_Defects/func_pointer.c:375	Not processed
 01.w_Defects/func_pointer.c:498	Not processed
 01.w_Defects/func_pointer.c:538	Not processed
 01.w_Defects/func_pointer.c:548	Not processed
 01.w_Defects/func_pointer.c:592	Not processed
 01.w_Defects/insign_code.c:24	Not processed
 01.w_Defects/invalid_memory_access.c:45	Not processed
 01.w_Defects/invalid_memory_access.c:147	Not processed
 01.w_Defects/invalid_memory_access.c:410	Not processed
 01.w_Defects/invalid_memory_access.c:422	Not processed
 01.w_Defects/invalid_memory_access.c:532	Not processed
 01.w_Defects/littlemem_st.c:36	Not processed
 01.w_Defects/lock_never_unlock.c:228	Not processed
 01.w_Defects/lock_never_unlock.c:325	Not processed
 01.w_Defects/memory_allocation_failure.c:282	Not processed
 01.w_Defects/memory_allocation_failure.c:448	Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Dead store

C/C++

 01.w_Defects/memory_leak.c:212	Not processed
 01.w_Defects/memory_leak.c:228	Not processed
 01.w_Defects/memory_leak.c:245	Not processed
 01.w_Defects/memory_leak.c:276	Not processed
 01.w_Defects/memory_leak.c:308	Not processed
 01.w_Defects/memory_leak.c:372	Not processed
 01.w_Defects/memory_leak.c:382	Not processed
 01.w_Defects/overrun_st.c:44	Not processed
 01.w_Defects/overrun_st.c:320	Not processed
 01.w_Defects/ptr_subtraction.c:35	Not processed
 01.w_Defects/race_condition.c:372	Not processed
 01.w_Defects/race_condition.c:392	Not processed
 01.w_Defects/st_underrun.c:137	Not processed
 01.w_Defects/st_underrun.c:195	Not processed
 01.w_Defects/st_underrun.c:229	Not processed
 01.w_Defects/underrun_st.c:21	Not processed
 01.w_Defects/underrun_st.c:55	Not processed
 01.w_Defects/uninit_memory_access.c:98	Not processed
 01.w_Defects/uninit_memory_access.c:169	Not processed
 01.w_Defects/uninit_memory_access.c:419	Not processed
 01.w_Defects/uninit_pointer.c:30	Not processed
 01.w_Defects/uninit_pointer.c:55	Not processed
 01.w_Defects/uninit_pointer.c:65	Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Dead store

C/C++

 01.w_Defects/uninit_pointer.c:81	Not processed
 01.w_Defects/uninit_pointer.c:90	Not processed
 01.w_Defects/uninit_pointer.c:200	Not processed
 01.w_Defects/uninit_pointer.c:335	Not processed
 01.w_Defects/uninit_var.c:23	Not processed
 01.w_Defects/uninit_var.c:34	Not processed
 01.w_Defects/uninit_var.c:45	Not processed
 01.w_Defects/uninit_var.c:63	Not processed
 01.w_Defects/uninit_var.c:81	Not processed
 01.w_Defects/uninit_var.c:111	Not processed
 01.w_Defects/uninit_var.c:186	Not processed
 01.w_Defects/uninit_var.c:228	Not processed
 01.w_Defects/uninit_var.c:249	Not processed
 01.w_Defects/uninit_var.c:276	Not processed
 01.w_Defects/uninit_var.c:296	Not processed
 01.w_Defects/unused_var.c:24	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:53	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:74	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:94	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:114	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:141	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:161	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:182	Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Dead store

C/C++

 01.w_Defects/wrong_arguments_func_pointer.c:201	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:225	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:282	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:423	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:491	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:502	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:528	Not processed
 01.w_Defects/wrong_arguments_func_pointer.c:539	Not processed
 01.w_Defects/zero_division.c:23	Not processed
 01.w_Defects/zero_division.c:35	Not processed
 01.w_Defects/zero_division.c:47	Not processed
 01.w_Defects/zero_division.c:78	Not processed
 01.w_Defects/zero_division.c:93	Not processed
 01.w_Defects/zero_division.c:118	Not processed
 01.w_Defects/zero_division.c:129	Not processed
 01.w_Defects/zero_division.c:141	Not processed
 01.w_Defects/zero_division.c:154	Not processed
 01.w_Defects/zero_division.c:166	Not processed
 01.w_Defects/zero_division.c:178	Not processed
 01.w_Defects/zero_division.c:195	Not processed
 01.w_Defects/zero_division.c:206	Not processed
 01.w_Defects/zero_division.c:225	Not processed
 01.w_Defects/zero_division.c:252	Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Dead store

C/C++

 02.wo_Defects/free_nondynamically_allocated_memory.c:60	Not processed
 02.wo_Defects/free_null_pointer.c:364	Not processed
 02.wo_Defects/free_null_pointer.c:425	Not processed
 02.wo_Defects/free_null_pointer.c:458	Not processed
 02.wo_Defects/func_pointer.c:341	Not processed
 02.wo_Defects/func_pointer.c:513	Not processed
 02.wo_Defects/func_pointer.c:565	Not processed
 02.wo_Defects/littlemem_st.c:37	Not processed
 02.wo_Defects/lock_never_unlock.c:231	Not processed
 02.wo_Defects/lock_never_unlock.c:327	Not processed
 02.wo_Defects/memory_allocation_failure.c:292	Not processed
 02.wo_Defects/memory_allocation_failure.c:457	Not processed
 02.wo_Defects/memory_leak.c:378	Not processed
 02.wo_Defects/race_condition.c:222	Not processed
 02.wo_Defects/race_condition.c:420	Not processed
 02.wo_Defects/race_condition.c:440	Not processed
 02.wo_Defects/uninit_memory_access.c:100	Not processed
 02.wo_Defects/uninit_memory_access.c:401	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:53	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:74	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:94	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:114	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:139	Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Dead store

C/C++

 02.wo_Defects/wrong_arguments_func_pointer.c:158	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:178	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:197	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:224	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:284	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:420	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:487	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:498	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:525	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:536	Not processed
 02.wo_Defects/zero_division.c:23	Not processed
 02.wo_Defects/zero_division.c:35	Not processed
 02.wo_Defects/zero_division.c:46	Not processed
 02.wo_Defects/zero_division.c:76	Not processed
 02.wo_Defects/zero_division.c:91	Not processed
 02.wo_Defects/zero_division.c:116	Not processed
 02.wo_Defects/zero_division.c:127	Not processed
 02.wo_Defects/zero_division.c:139	Not processed
 02.wo_Defects/zero_division.c:154	Not processed
 02.wo_Defects/zero_division.c:167	Not processed
 02.wo_Defects/zero_division.c:179	Not processed
 02.wo_Defects/zero_division.c:196	Not processed
 02.wo_Defects/zero_division.c:207	Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Dead store

C/C++

 02.wo_Defects/zero_division.c:226	Not processed
 02.wo_Defects/zero_division.c:253	Not processed
 03.w_Defects_Cpp/improper_error_handling.cpp:22	Not processed
 03.w_Defects_Cpp/improper_error_handling.cpp:42	Not processed
 03.w_Defects_Cpp/improper_error_handling.cpp:63	Not processed
 03.w_Defects_Cpp/improper_error_handling.cpp:87	Not processed
 04.wo_Defects_Cpp/improper_error_handling.cpp:23	Not processed
 04.wo_Defects_Cpp/improper_error_handling.cpp:44	Not processed
 04.wo_Defects_Cpp/improper_error_handling.cpp:66	Not processed
 04.wo_Defects_Cpp/improper_error_handling.cpp:91	Not processed

Garbage memory usage is possible

C/C++

15

 01.w_Defects/littlemem_st.c:93	Not processed
 01.w_Defects/return_local.c:19	Not processed
 01.w_Defects/st_cross_thread_access.c:55	Not processed
 01.w_Defects/st_cross_thread_access.c:143	Not processed
 01.w_Defects/st_cross_thread_access.c:231	Not processed
 01.w_Defects/st_cross_thread_access.c:320	Not processed
 01.w_Defects/st_cross_thread_access.c:399	Not processed
 01.w_Defects/st_cross_thread_access.c:482	Not processed
 02.wo_Defects/littlemem_st.c:94	Not processed
 02.wo_Defects/st_cross_thread_access.c:56	Not processed
 02.wo_Defects/st_cross_thread_access.c:144	Not processed
 02.wo_Defects/st_cross_thread_access.c:233	Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Garbage memory usage is possible

C/C++

- 02.wo_Defects/st_cross_thread_access.c:321
- 02.wo_Defects/st_cross_thread_access.c:400
- 02.wo_Defects/st_cross_thread_access.c:483

Not processed
Not processed
Not processed

Incorrect function call

C/C++

13

- 01.w_Defects/func_pointer.c:375
- 01.w_Defects/memory_allocation_failure.c:514
- 01.w_Defects/uninit_memory_access.c:27
- 01.w_Defects/uninit_memory_access.c:54
- 01.w_Defects/uninit_memory_access.c:127
- 01.w_Defects/uninit_memory_access.c:319
- 01.w_Defects/uninit_pointer.c:71
- 01.w_Defects/uninit_pointer.c:187
- 01.w_Defects/uninit_pointer.c:358
- 01.w_Defects/wrong_arguments_func_pointer.c:161
- 01.w_Defects/wrong_arguments_func_pointer.c:381
- 01.w_Defects/wrong_arguments_func_pointer.c:423
- 01.w_Defects/wrong_arguments_func_pointer.c:457

Not processed
Not processed

Use of overlapping buffers

C/C++

1

- 01.w_Defects/wrong_arguments_func_pointer.c:213

Not processed

Using an insecure method

C/C++

40

- 01.w_Defects/deletion_of_data_structure_sentinel.c:41
- 01.w_Defects/free_null_pointer.c:108

Not processed
Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Using an insecure method

C/C++

 01.w_Defects/free_null_pointer.c:240	Not processed
 01.w_Defects/function_return_value_unchecked.c:334	Not processed
 01.w_Defects/func_pointer.c:97	Not processed
 01.w_Defects/func_pointer.c:391	Not processed
 01.w_Defects/invalid_memory_access.c:505	Not processed
 01.w_Defects/main.c:22	Not processed
 01.w_Defects/memory_leak.c:72	Not processed
 01.w_Defects/memory_leak.c:399	Not processed
 01.w_Defects/memory_leak.c:423	Not processed
 01.w_Defects/null_pointer.c:237	Not processed
 01.w_Defects/st_underrun.c:24	Not processed
 01.w_Defects/st_underrun.c:50	Not processed
 01.w_Defects/st_underrun.c:85	Not processed
 01.w_Defects/st_underrun.c:122	Not processed
 01.w_Defects/st_underrun.c:191	Not processed
 01.w_Defects/st_underrun.c:225	Not processed
 02.wo_Defects/buffer_underrun_dynamic.c:647	Not processed
 02.wo_Defects/deletion_of_data_structure_sentinel.c:42	Not processed
 02.wo_Defects/free_null_pointer.c:118	Not processed
 02.wo_Defects/free_null_pointer.c:248	Not processed
 02.wo_Defects/function_return_value_unchecked.c:335	Not processed
 02.wo_Defects/func_pointer.c:104	Not processed
 02.wo_Defects/func_pointer.c:407	Not processed

* Rejected vulnerabilities are not taken into account

Low-level vulnerabilities

Using an insecure method

C/C++

 02.wo_Defects/invalid_memory_access.c:510	Not processed
 02.wo_Defects/main.c:22	Not processed
 02.wo_Defects/memory_allocation_failure.c:224	Not processed
 02.wo_Defects/memory_leak.c:75	Not processed
 02.wo_Defects/memory_leak.c:405	Not processed
 02.wo_Defects/memory_leak.c:430	Not processed
 02.wo_Defects/null_pointer.c:258	Not processed
 02.wo_Defects/st_underrun.c:25	Not processed
 02.wo_Defects/st_underrun.c:51	Not processed
 02.wo_Defects/st_underrun.c:86	Not processed
 02.wo_Defects/st_underrun.c:123	Not processed
 02.wo_Defects/st_underrun.c:234	Not processed
 02.wo_Defects/wrong_arguments_func_pointer.c:380	Not processed
 03.w_Defects_Cpp/main.cpp:24	Not processed
 04.wo_Defects_Cpp/main.cpp:23	Not processed

* Rejected vulnerabilities are not taken into account

Detailed Results

Buffer overflow (C/C++)

Description

The program contains a possible buffer overflow. An attacker is able to overrun the buffer's boundary and execute an arbitrary code or perform a DoS attack on the system.

Example

The following example shows a potential buffer overflow.

```
int main(int argc, char **argv) {
    char buf[256];
    strcpy(buf, argv[0]);
    printf("%s\n", buf);
    return 0;
}
```

A safe alternative.

```
int main(int argc, char **argv) {
    char buf[256];
    strncpy(buf, argv[0], 256);
    printf("%s\n", buf);
    return 0;
}
```

Recommendations

- Check buffer boundaries before read and write operations on buffer.

Links

1. Buffer Overflow Exploit — Dhaval Kapil

Vulnerability Entries

01.w_Defects/st_underrun.c:204

Level Critical**Status** Not processed

```
201 void st_underrun_006 ()  
202 {  
203     st_underrun_006_s_001 s;  
  
204     strcpy(s.buf,"STRING !!!!");  
  
205     void (*func)(st_underrun_006_s_001);  
206     func = st_underrun_006_func_001;  
207     func(s);
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;  
261 void st_underrun_main ()  
262 {  
  
263     if (vflag == 1 || vflag ==888)  
  
264     {  
265         st_underrun_001();  
266     }
```

strcpy(s.buf,"STRING !!!")

01.w_Defects/st_underrun.c:204

```
201 void st_underrun_006 ()  
202 {  
203     st_underrun_006_s_001 s;  
  
204     strcpy(s.buf,"STRING !!!");  
  
205     void (*func)(st_underrun_006_s_001);
```

```
206 func = st_underrun_006_func_001;
207 func(s);
```

02.wo_Defects/st_underrun.c:213

Level Critical

Status Not processed

```
210 // JDR: this is an array overrun (copying 12 bytes into a 10-byte array)
211 // but maybe it is OK since the struct is guaranteed to have buf1 right
212 // after buf?
```

213 strcpy(s.buf,"STRING !!!!");

```
214 void (*func)(st_underrun_006_s_001);
215 func = st_underrun_006_func_001;
216 func(s);
```

Trace

vflag == 1

02.wo_Defects/st_underrun.c:273

```
270 extern volatile int vflag;
271 void st_underrun_main ()
272 {
```

273 if (vflag == 1 || vflag ==888)

```
274 {
275     st_underrun_001();
276 }
```

```
strcpy(s.buf,"STRING !!!!")
```

02.wo_Defects/st_underrun.c:213

```
210     // JDR: this is an array overrun (copying 12 bytes into a 10-byte array)
211     // but maybe it is OK since the struct is guaranteed to have buf1 right
212     // after buf?

213 strcpy(s.buf,"STRING !!!");

214 void (*func)(st_underrun_006_s_001);
215 func = st_underrun_006_func_001;
216 func(s);
```

Dangerous pointer manipulation (C/C++)

Description

Some arithmetic manipulations on pointers may be dangerous:

1. Pointer arithmetic on a pointer to base class is dangerous - base and derived may have different sizes.
2. Pointer arithmetic on non-array variables relies on memory layout, which is dangerous.
3. Subtraction of two pointers that do not point to the same memory chunk may cause incorrect result.

Example

An example of when a vulnerability arises as a result of arithmetic operations with a pointer to the base class:

```
class Base {};
class Derived : public Base {};
void checkPolymorphicUse() {
    Derived d[10];
    Base *p = d;
    ++p;
}
```

An example of when a vulnerability is associated with the arithmetic of pointers to variables that are not part of an array:

```
void checkBasicArithmetic(int i) {
    int t[10];
    int *p = t;
```

```
++p;  
int a = 5;  
p = &a;  
++p;  
}
```

An example where the subtraction of two pointers pointing to the same memory fragment occurs:

```
void badFunction() {  
    int x, y;  
    int d = &y - &x;  
}
```

Recommendations

- Platforms with high-level memory abstractions should be used.
- Always use array indexing instead of direct pointer manipulation.
- Use buffer overflow prevention technologies.

Links

1. C-Pointer arithmetic
2. CWE-468: Incorrect Pointer Scaling

Vulnerability Entries

01.w_Defects/ptr_subtraction.c:22

Level Critical

Status Not processed

```
19 char buf1[5];  
20 char buf2[5];  
21 intptr_t offset;
```

```
22 offset = buf2 - buf1; /*Tool should detect this line as error*/ /*ERROR:Incorrect  
pointer arithmetic*/
```

```
23     sink = offset;  
24 }  
25
```

Trace

vflag ==1

01.w_Defects/ptr_subtraction.c:45

```
42 extern volatile int vflag;
43 void ptr_subtraction_main ()
44 {
45     if (vflag ==1 || vflag ==888)
46     {
47         ptr_subtraction_001();
48     }
```

buf2 - buf1

01.w_Defects/ptr_subtraction.c:22

```
19     char buf1[5];
20     char buf2[5];
21     intptr_t offset;
22     offset = buf2 - buf1; /*Tool should detect this line as error*/ /*ERROR:
Incorrect pointer arithmetic*/
23     sink = offset;
24 }
25
```

01.w_Defects/ptr_subtraction.c:35

Level Critical

Status Not processed

```
32     int x= 10;
33     int *ptr = &x;
34     char *buf ;
```

35 buf= (char *)(ptr+1); /*Tool should detect this line as error*/ /*ERROR:Incorrect

```
pointer arithmetic*/
```

```
36 }  
37  
38 /*
```

Trace

```
vflag ==1
```

```
01.w_Defects/ptr_subtraction.c:45
```

```
42 extern volatile int vflag;  
43 void ptr_subtraction_main ()  
44 {  
  
45     if (vflag ==1 || vflag ==888)  
  
46     {  
47         ptr_subtraction_001();  
48     }
```

```
buf= (char *)(ptr+1); /*Tool should detect this  
line as error*/ /*ERROR:Incorrect pointer  
arithmetic*/
```

```
01.w_Defects/ptr_subtraction.c:35
```

```
32     int x= 10;  
33     int *ptr = &x;  
34     char *buf ;  
  
35     buf= (char *)(ptr+1); /*Tool should detect this line as error*/ /*ERROR:  
Incorrect pointer arithmetic*/  
  
36 }  
37  
38 /*
```

02.wo_Defects/ptr_subtraction.c:23

Level Critical**Status** Not processed

```
20 int buf1[10];
21 int buf2[5];
22 intptr_t offset;
```

```
23 offset = buf1 - buf2; /*Tool should not detect this line as error*/ /*No ERROR:
Incorrect pointer arithmetic*/
```

```
24     sink = offset;
25 }
26
```

Trace

vflag ==1

02.wo_Defects/ptr_subtraction.c:46

```
43 extern volatile int vflag;
44 void ptr_subtraction_main ()
45 {
46     if (vflag ==1 || vflag ==888)
47     {
48         ptr_subtraction_001();
49     }
```

buf1 - buf2

02.wo_Defects/ptr_subtraction.c:23

```
20 int buf1[10];
21 int buf2[5];
22 intptr_t offset;
```

```
23 offset = buf1 - buf2; /*Tool should not detect this line as error*/ /*No ERROR:
Incorrect pointer arithmetic*/
```

```
24     sink = offset;
25 }
26
```

Division by zero (C/C++)

Description

Division by zero may lead to malfunction or crash of the application.

Division by zero often occurs when calculating values of the variables that store length, width, height of some object. If the value of the argument that eventually causes division by zero is obtained from an untrusted source, an attacker can disrupt the normal operation of the application.

Example

An example of a possible division by zero:
return (a % (qX-1));

Recommendations

- Check the value of the denominator for equality to zero before dividing.
- If the denominator is of float type, instead of comparing for equality you should check whether its value lies in a certain small range around zero. It is incorrect to check floating point values for equality.

Links

1. CWE-369: Divide By Zero

Vulnerability Entries

01.w_Defects/zero_division.c:23

Level Medium**Status** Not processed

```
20 {  
21     int dividend = 1000;  
22     int ret;  
  
23     ret = dividend / 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
24 }  
25  
26 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {  
  
262     if (vflag == 1 || vflag == 888)  
  
263     {  
264         zero_division_001();  
265     }
```

dividend / 0

01.w_Defects/zero_division.c:23

```
20 {  
21     int dividend = 1000;  
22     int ret;  
  
23     ret = dividend / 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */
```

```
24 }  
25  
26 /*
```

01.w_Defects/zero_division.c:34

Level Medium**Status** Not processed

```
31 {  
32     int dividend = 1000;  
33     int ret;  
  
34     dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
35     ret = dividend;  
36 }  
37
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {  
  
262     if (vflag == 1 || vflag == 888)  
  
263     {  
264         zero_division_001();  
265     }
```

```
dividend /= 0
```

01.w_Defects/zero_division.c:34

```
31 {  
32     int dividend = 1000;  
33     int ret;  
  
34     dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by  
zero */  
  
35     ret = dividend;  
36 }  
37
```

01.w_Defects/zero_division.c:47

Level Medium

Status Not processed

```
44 {  
45     int dividend = 1000;  
46     int ret;  
  
47     ret = dividend % 0; /*Tool should detect this line as error*/ /* ERROR:division by  
zero */  
  
48 }  
49  
50
```

Trace

```
vflag == 1
```

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

```
262 if (vflag == 1 || vflag ==888)
```

```
263 {  
264     zero_division_001();  
265 }
```

dividend % 0

01.w_Defects/zero_division.c:47

```
44 {  
45     int dividend = 1000;  
46     int ret;
```

```
47     ret = dividend % 0; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */
```

```
48 }  
49  
50
```

01.w_Defects/zero_division.c:78

Level Medium

Status Not processed

```
75     int dividend = 1000;  
76     int divisors[5] = {2, 1, 0, 3, 4};  
77     int ret;
```

```
78     ret = dividend / divisors[2]; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */
```

```
79 }  
80  
81 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;
260 void zero_division_main ()
261 {

262     if (vflag == 1 || vflag == 888)

263     {
264         zero_division_001();
265     }
```

dividend / divisors[2]

01.w_Defects/zero_division.c:78

```
75     int dividend = 1000;
76     int divisors[5] = {2, 1, 0, 3, 4};
77     int ret;

78     ret = dividend / divisors[2]; /*Tool should detect this line as error*/ /* ERROR:
division by zero */

79 }
80
81 /*
```

01.w_Defects/zero_division.c:118

Level Medium

Status Not processed

```
115     int dividend = 1000;
116     int ret;
117     zero_division_007_func_001();
```

```
118     ret = dividend / zero_division_007_s_gbl.divisor; /*Tool should detect this line as
error*/ /* ERROR: division by zero */
```

```
119 }  
120  
121 /*
```

Trace

```
vflag == 1
```

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

```
262 if (vflag == 1 || vflag ==888)
```

```
263 {  
264     zero_division_001();  
265 }
```

```
dividend / zero_division_007_s_gbl.divisor
```

01.w_Defects/zero_division.c:118

```
115 int dividend = 1000;  
116 int ret;  
117 zero_division_007_func_001();
```

```
118 ret = dividend / zero_division_007_s_gbl.divisor; /*Tool should detect this line  
as error*/ /* ERROR:division by zero */
```

```
119 }  
120  
121 /*
```

01.w_Defects/zero_division.c:141

Level Medium

Status Not processed

```
138 int dividend = 1000;
139 int divisor = 0;
140 int ret;

141 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:division by
zero */

142 }
143
144 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;
260 void zero_division_main ()
261 {

262 if (vflag == 1 || vflag ==888)

263 {
264     zero_division_001();
265 }
```

dividend / divisor

01.w_Defects/zero_division.c:141

```
138 int dividend = 1000;
139 int divisor = 0;
140 int ret;

141 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:
division by zero */

142 }
143
144 /*
```

01.w_Defects/zero_division.c:166

Level Medium**Status** Not processed

```
163 int dividend = 1000;  
164 int divisor = 2;  
165 int ret;
```

```
166 ret = dividend / (2 * divisor - 4);/*Tool should detect this line as error*/ /* ERROR:  
division by zero */
```

```
167 }  
168  
169 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {  
  
262 if (vflag == 1 || vflag ==888)  
  
263 {  
264     zero_division_001();  
265 }
```

dividend / (2 * divisor - 4)

01.w_Defects/zero_division.c:166

```
163 int dividend = 1000;  
164 int divisor = 2;  
165 int ret;
```

```
166 ret = dividend / (2 * divisor - 4);/*Tool should detect this line as error*/ /*  
ERROR:division by zero */
```

```
167 }  
168  
169 /*
```

01.w_Defects/zero_division.c:178

Level Medium

Status Not processed

```
175 int dividend = 1000;  
176 int divisor = 2;  
177 int ret;
```

178 **ret = dividend / (divisor * divisor - 4);/*Tool should detect this line as error*/ /***
ERROR:division by zero */

```
179  
180 }  
181
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

262 if (vflag == 1 || vflag == 888)

```
263 {  
264     zero_division_001();  
265 }
```

```
dividend / (divisor * divisor - 4)
```

01.w_Defects/zero_division.c:178

```
175 int dividend = 1000;  
176 int divisor = 2;  
177 int ret;
```

```
178 ret = dividend / (divisor * divisor - 4);/*Tool should detect this line as error*/ /*  
ERROR:division by zero */
```

```
179  
180 }  
181
```

01.w_Defects/zero_division.c:195

Level Medium

Status Not processed

```
192 {  
193 int dividend = 1000;  
194 int ret;
```

```
195 ret = dividend / zero_division_013_func_001();/*Tool should detect this line as  
error*/ /* ERROR:division by zero */
```

```
196 }  
197  
198 /*
```

Trace

```
vflag == 1
```

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

```
262 if (vflag == 1 || vflag ==888)
```

```
263 {  
264     zero_division_001();  
265 }
```

```
dividend / zero_division_013_func_001()
```

01.w_Defects/zero_division.c:195

```
192 {  
193     int dividend = 1000;  
194     int ret;  
  
195     ret = dividend / zero_division_013_func_001();/*Tool should detect this line  
as error*/ /* ERROR:division by zero */  
  
196 }  
197  
198 /*
```

01.w_Defects/zero_division.c:206

Level Medium

Status Not processed

```
203 {  
204     int dividend = 1000;  
205     int ret;
```

```
206     ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:division by  
zero */
```

```
207 }  
208  
209 void zero_division_014 ()
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;
260 void zero_division_main ()
261 {

262 if (vflag == 1 || vflag ==888)

263 {
264     zero_division_001();
265 }
```

dividend / divisor

01.w_Defects/zero_division.c:206

```
203 {
204     int dividend = 1000;
205     int ret;

206     ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:
division by zero */

207 }
208
209 void zero_division_014 ()
```

01.w_Defects/zero_division.c:225

Level Medium

Status Not processed

```
222 int divisor1;
223 int ret;
224 divisor1 = divisor;
```

```
225 ret = dividend / divisor1; /*Tool should detect this line as error*/ /* ERROR:division
by zero */
```

```
226 }  
227  
228 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {  
  
262     if (vflag == 1 || vflag == 888)  
  
263     {  
264         zero_division_001();  
265     }
```

dividend / divisor1

01.w_Defects/zero_division.c:225

```
222     int divisor1;  
223     int ret;  
224     divisor1 = divisor;  
  
225     ret = dividend / divisor1; /*Tool should detect this line as error*/ /* ERROR:  
division by zero */  
  
226 }  
227  
228 /*
```

01.w_Defects/zero_division.c:252

Level Medium

Status Not processed

```
249 zero_division_016_func_002();  
250 divisor1 = *zero_division_016_gbl_divisor;  
251 divisor2 = divisor1;  
  
252 ret = dividend / divisor2; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */  
  
253 }  
254  
255 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {  
  
262 if (vflag == 1 || vflag == 888)  
  
263 {  
264     zero_division_001();  
265 }
```

dividend / divisor2

01.w_Defects/zero_division.c:252

```
249 zero_division_016_func_002();  
250 divisor1 = *zero_division_016_gbl_divisor;  
251 divisor2 = divisor1;  
  
252 ret = dividend / divisor2; /*Tool should detect this line as error*/ /* ERROR:  
division by zero */  
  
253 }  
254  
255 /*
```

03.w_Defects_Cpp/improper_error_handling.cpp:22

Level Medium**Status** Not processed

```
19 try {  
20     int a=0,b=9,c;  
21     if (a==0)  
  
22     c=b/a;  
  
23 }  
24 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error  
handling*/  
25 {
```

Trace

vflag == 1

03.w_Defects_Cpp/improper_error_handling.cpp:108

```
105 extern volatile int vflag;  
106 void improper_error_handling_main ()  
107 {  
  
108     if (vflag == 1 || vflag ==888)  
  
109     {  
110         improper_error_handling_001();  
111     }
```

b/a

03.w_Defects_Cpp/improper_error_handling.cpp:22

```
19 try {  
20     int a=0,b=9,c;  
21     if (a==0)  
  
22     c=b/a;
```

```
23 }  
24 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error  
handling*/  
25 {
```

Double free (C/C++)

Description

The application is calling memory deallocation function twice on the same value. This may lead to memory leak or undefined behavior of the application.

When a program calls `free()` twice with the same argument, the program's memory management data structures become corrupted and could allow a malicious user to write values in arbitrary memory spaces. This corruption can cause the program to crash or, in some circumstances, alter the execution flow.

Example

Example of a double free:

```
char* ptr = (char*)malloc (SIZE);
```

```
...  
if (abrt) {  
    free(ptr);  
}  
...  
free(ptr);
```

Recommendations

- Ensure that each allocation is freed only once.
- After freeing a chunk, set the pointer to `NULL` to ensure the pointer cannot be freed again.

Links

1. Double Free — owasp.org
2. CWE-415: Double Free

Vulnerability Entries

01.w_Defects/double_free.c:22

Level Medium

Status Not processed

```
19 char* ptr= (char*) malloc(sizeof(char));
20 free(ptr);
21
22 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
23 }
24
25 /*
```

Trace

vflag == 1

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {
233     if (vflag == 1 || vflag ==888)
234     {
235         double_free_001 ();
236     }
```

```
free(ptr)
```

01.w_Defects/double_free.c:22

```
19 char* ptr= (char*) malloc(sizeof(char));
20 free(ptr);
21

22 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/

23 }
24
25 /*
```

01.w_Defects/double_free.c:43

Level Medium

Status Not processed

```
40             free(ptr);
41         }
42     }

43 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/

44 }
45
46 /*
```

Trace

```
vflag == 1
```

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {

233 if (vflag == 1 || vflag ==888)
```

```
234 {  
235     double_free_001 ();  
236 }
```

free(ptr)

01.w_Defects/double_free.c:43

```
40         free(ptr);  
41     }  
42 }  
  
43 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/  
  
44 }  
45  
46 /*
```

01.w_Defects/double_free.c:64

Level Medium

Status Not processed

```
61         free(ptr);  
62     }  
63 }  
  
64 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/  
  
65 }  
66  
67 /*
```

Trace

```
vflag == 1
```

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {
233     if (vflag == 1 || vflag == 888)
234     {
235         double_free_001 ();
236     }
}
```

```
free(ptr)
```

01.w_Defects/double_free.c:64

```
61             free(ptr);
62         }
63     }

64     free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
65 }
66
67 /*
```

01.w_Defects/double_free.c:87

Level Medium

Status Not processed

```
84 }
85
86     if(rand() % 3 == 0)

87     free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
88 }
```

```
89
90 /*
```

Trace

```
vflag == 1
```

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {
233     if (vflag == 1 || vflag ==888)
234     {
235         double_free_001 ();
236     }
```

```
free(ptr)
```

01.w_Defects/double_free.c:87

```
84 }
85
86 if(rand() % 3==0)
87 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
88 }
89
90 /*
```

01.w_Defects/double_free.c:101

Level Medium

Status Not processed

```
98     free(ptr);
```

```
99
100 if(ptr)

101 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/

102 }
103
104 /*
```

Trace

vflag == 1

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {

233 if (vflag == 1 || vflag ==888)

234 {
235     double_free_001 ();
236 }
```

free(ptr)

01.w_Defects/double_free.c:101

```
98 free(ptr);
99
100 if(ptr)

101 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/

102 }
103
104 /*
```

01.w_Defects/double_free.c:115

Level Medium**Status** Not processed

```
112 if(1)
113 free(ptr);
114

115 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/

116 }
117
118 /*
```

Trace

vflag == 1

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {

233 if (vflag == 1 || vflag ==888)

234 {
235     double_free_001 ();
236 }
```

free(ptr)

01.w_Defects/double_free.c:115

```
112 if(1)
113 free(ptr);
114

115 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/

116 }
```

```
117  
118 /*
```

01.w_Defects/double_free.c:131

Level Medium

Status Not processed

```
128 if(flag>=0)  
129 free(ptr);  
130
```

131 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/

```
132 }  
133  
134 /*
```

Trace

vflag == 1

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;  
231 void double_free_main ()  
232 {
```

233 if (vflag == 1 || vflag ==888)

```
234 {  
235     double_free_001 ();  
236 }
```

```
free(ptr)
```

01.w_Defects/double_free.c:131

```
128 if(flag>=0)
129 free(ptr);
130

131 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/

132 }
133
134 /*
```

01.w_Defects/double_free.c:149

Level Medium

Status Not processed

```
146 double_free_function_008_gbl_ptr= (char*) malloc(sizeof(char));
147
148 double_free_function_008();

149 free(double_free_function_008_gbl_ptr); /*Tool should detect this line as error*/
/*ERROR:Double free*/

150 }
151
152 /*
```

Trace

```
vflag == 1
```

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {
```

```
233 if (vflag == 1 || vflag ==888)
```

```
234 {  
235     double_free_001 ();  
236 }
```

```
free(double_free_function_008_gbl_ptr)
```

01.w_Defects/double_free.c:149

```
146 double_free_function_008_gbl_ptr= (char*) malloc(sizeof(char));  
147  
148 double_free_function_008();
```

```
149 free(double_free_function_008_gbl_ptr); /*Tool should detect this line as  
error*/ /*ERROR:Double free*/
```

```
150 }  
151  
152 /*
```

01.w_Defects/double_free.c:168

Level Medium

Status Not processed

```
165     free(ptr);  
166     flag++;  
167 }
```

```
168 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
```

```
169 }  
170  
171 /*
```

Trace

```
vflag == 1
```

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {
233     if (vflag == 1 || vflag == 888)
234     {
235         double_free_001 ();
236     }
```

```
free(ptr)
```

01.w_Defects/double_free.c:168

```
165     free(ptr);
166     flag++;
167 }
168 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
169 }
170
171 /*
```

01.w_Defects/double_free.c:187

Level Medium

Status Not processed

```
184     free(ptr);
185     flag--;
186 }
```

187 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/

```
188 }
```

```
189  
190 /*
```

Trace

```
vflag == 1
```

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;  
231 void double_free_main ()  
232 {  
  
233     if (vflag == 1 || vflag ==888)  
  
234     {  
235         double_free_001 ();  
236     }
```

```
free(ptr)
```

01.w_Defects/double_free.c:187

```
184     free(ptr);  
185     flag--;  
186 }  
  
187 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/  
  
188 }  
189  
190 /*
```

01.w_Defects/double_free.c:204

Level Medium

Status Not processed

```
201 while(a<b)
```

```
202 {  
203     if(flag ==1)  
  
204         free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/  
  
205     a++;  
206 }  
207 }
```

Trace

vflag == 1

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;  
231 void double_free_main ()  
232 {  
  
233     if (vflag == 1 || vflag ==888)  
  
234     {  
235         double_free_001 ();  
236     }
```

free(ptr)

01.w_Defects/double_free.c:204

```
201 while(a<b)  
202 {  
203     if(flag ==1)  
  
204         free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double  
free*/  
  
205     a++;  
206 }  
207 }
```

01.w_Defects/double_free.c:222

Level Medium**Status** Not processed

```
219
220 for(a=0;a<2;a++)
221 {
222     free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
223 }
224 }
225
```

Trace

vflag == 1

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {
233     if (vflag == 1 || vflag ==888)
234     {
235         double_free_001 ();
236     }
```

free(ptr)

01.w_Defects/double_free.c:222

```
219
220 for(a=0;a<2;a++)
221 {
222     free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double
free*/
```

```
223 }
224 }
225
```

Function calling with invalid null argument (C/C++)

Description

The function is called with invalid null argument. This may lead to incorrect behavior of the application.

The function is used with arguments whose values should not be NULL, however, checking for null is not performed. Undefined behavior of the application is possible.

Example

In the following example, the function memcpy() has a null pointer as argument instead a pointer to the source, which causes undefined behavior of the application:
memcpy (dst, NULL, 6);

Recommendations

- Check the argument for equality to null before calling corresponding function.

Links

1. Is it guaranteed to be safe to perform memcpy(0,0,0)? — stackoverflow.com

Vulnerability Entries

01.w_Defects/free_null_pointer.c:109

Level Medium

Status Not processed

```
106 {  
107     char *str = "This is a string";  
108     free_null_pointer_005_func_001(strlen(str));  
  
109     strcpy(free_null_pointer_005_gbl_ptr,str);  
  
110     free(free_null_pointer_005_gbl_ptr); /* Tool should detect this line as error  
*///*ERROR:Freeing a NULL pointer*/  
111     free_null_pointer_005_gbl_ptr = NULL;  
112 }
```

Trace

vflag == 1

01.w_Defects/free_null_pointer.c:570

```
567 extern volatile int vflag;  
568 void free_null_pointer_main ()  
569 {  
  
570     if (vflag == 1 || vflag ==888)  
  
571     {  
572         free_null_pointer_001();  
573     }
```

strcpy(free_null_pointer_005_gbl_ptr,str)

01.w_Defects/free_null_pointer.c:109

```
106 {  
107     char *str = "This is a string";  
108     free_null_pointer_005_func_001(strlen(str));  
  
109     strcpy(free_null_pointer_005_gbl_ptr,str);
```

```
110 free(free_null_pointer_005_gbl_ptr); /* Tool should detect this line as error
*///*ERROR:Freeing a NULL pointer*/
111 free_null_pointer_005_gbl_ptr = NULL;
112 }
```

01.w_Defects/free_null_pointer.c:146

Level Medium**Status** Not processed

```
143 {
144     (s+i)->buf = NULL;
145 }

146 strcpy((s+4)->buf,s1);

147 }
148 if(free_null_pointer_006_func_001(flag)==0)
149 {
```

Trace

vflag == 1

01.w_Defects/free_null_pointer.c:570

```
567 extern volatile int vflag;
568 void free_null_pointer_main ()
569 {

570 if (vflag == 1 || vflag ==888)

571 {
572     free_null_pointer_001();
573 }
```

```
strcpy((s+4)->buf,s1)
```

01.w_Defects/free_null_pointer.c:146

```
143 {  
144     (s+i)->buf = NULL;  
145 }  
  
146 strcpy((s+4)->buf,s1);  
  
147 }  
148 if(free_null_pointer_006_func_001(flag)==0)  
149 {
```

01.w_Defects/free_null_pointer.c:241

Level Medium

Status Not processed

```
238 char *str = "This is a string";  
239 char *str1=NULL;  
240 free_null_pointer_008_func_001(strlen(str),&str1);  
  
241 strcpy(str1,str);  
  
242 free(str1);/* Tool should detect this line as error *//*ERROR:Freeing a NULL  
pointer*/  
243 str1 = NULL;  
244 }
```

Trace

```
vflag == 1
```

01.w_Defects/free_null_pointer.c:570

```
567 extern volatile int vflag;  
568 void free_null_pointer_main ()  
569 {
```

```
570 if (vflag == 1 || vflag ==888)

571 {
572     free_null_pointer_001();
573 }
```

strcpy(str1,str)

01.w_Defects/free_null_pointer.c:241

```
238 char *str = "This is a string";
239 char *str1=NULL;
240 free_null_pointer_008_func_001(strlen(str),&str1);

241 strcpy(str1,str);

242 free(str1);/* Tool should detect this line as error *//*ERROR:Freeing a NULL
pointer*/
243 str1 = NULL;
244 }
```

01.w_Defects/memory_leak.c:424

Level Medium

Status Not processed

```
421 {
422 char *str = "This is a string";
423 memory_leak_0016_func_001(strlen(str));

424 strcpy(memory_leak_0016_gbl_ptr,str);

425 }
426
427 /*
```

Trace

```
vflag == 1
```

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;
537 void memory_leak_main ()
538 {
539   if (vflag == 1 || vflag ==888)
540   {
541     memory_leak_001();
542 }
```

```
strcpy(memory_leak_0016_gbl_ptr,str)
```

01.w_Defects/memory_leak.c:424

```
421 {
422   char *str = "This is a string";
423   memory_leak_0016_func_001(strlen(str));
424   strcpy(memory_leak_0016_gbl_ptr,str);
425 }
426
427 /*
```

01.w_Defects/null_pointer.c:238

Level Medium

Status Not processed

```
235 {
236   char *str = "This is a string";
237   null_pointer_015_func_001(strlen(str));
```

```
238   strcpy(null_pointer_015_gbl_ptr,str);/*Tool should detect this line as error*/
/*ERROR:NULL pointer dereference*/
```

```
239 free(null_pointer_015_gbl_ptr);
240 null_pointer_015_gbl_ptr = NULL;
241 }
```

Trace

```
vflag == 1
```

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;
354 void null_pointer_main ()
355 {
356     if (vflag == 1 || vflag ==888)
357     {
358         null_pointer_001();
359     }
}
```

```
strcpy(null_pointer_015_gbl_ptr,str)
```

01.w_Defects/null_pointer.c:238

```
235 {
236     char *str = "This is a string";
237     null_pointer_015_func_001(strlen(str));
238     strcpy(null_pointer_015_gbl_ptr,str);/*Tool should detect this line as error*/
/*ERROR:NULL pointer dereference*/
239     free(null_pointer_015_gbl_ptr);
240     null_pointer_015_gbl_ptr = NULL;
241 }
```

02.wo_Defects/free_null_pointer.c:119

Level Medium

Status Not processed

```
116 {  
117     char *str = "This is a string";  
118     free_null_pointer_005_func_001(strlen(str));  
  
119     strcpy(free_null_pointer_005_gbl_ptr,str);  
  
120     free(free_null_pointer_005_gbl_ptr);/* Tool should not detect this line as error */  
/*No ERROR:Freeing a NULL pointer*/  
121     free_null_pointer_005_gbl_ptr = NULL;  
122 }
```

Trace

```
vflag == 1
```

```
02.wo_Defects/free_null_pointer.c:572
```

```
569 extern volatile int vflag;  
570 void free_null_pointer_main ()  
571 {  
  
572     if (vflag == 1 || vflag ==888)  
  
573     {  
574         free_null_pointer_001();  
575     }
```

```
strcpy(free_null_pointer_005_gbl_ptr,str)
```

```
02.wo_Defects/free_null_pointer.c:119
```

```
116 {  
117     char *str = "This is a string";  
118     free_null_pointer_005_func_001(strlen(str));  
  
119     strcpy(free_null_pointer_005_gbl_ptr,str);  
  
120     free(free_null_pointer_005_gbl_ptr);/* Tool should not detect this line as  
error */ /*No ERROR:Freeing a NULL pointer*/  
121     free_null_pointer_005_gbl_ptr = NULL;  
122 }
```

02.wo_Defects/free_null_pointer.c:249

Level Medium

Status Not processed

```
246 char *str = "This is a string";
247 char *str1=NULL;
248 free_null_pointer_008_func_001(strlen(str),&str1);

249 strcpy(str1,str);

250 free(str1);/* Tool should not detect this line as error */ /*No ERROR:Freeing a NULL
pointer*/
251 str1 = NULL;
252 }
```

Trace

vflag == 1

02.wo_Defects/free_null_pointer.c:572

```
569 extern volatile int vflag;
570 void free_null_pointer_main ()
571 {

572 if (vflag == 1 || vflag ==888)

573 {
574     free_null_pointer_001();
575 }
```

strcpy(str1,str)

02.wo_Defects/free_null_pointer.c:249

```
246 char *str = "This is a string";
247 char *str1=NULL;
248 free_null_pointer_008_func_001(strlen(str),&str1);

249 strcpy(str1,str);
```

```
250 free(str1);/* Tool should not detect this line as error */ /*No ERROR:Freeing  
a NULL pointer*/  
251 str1 = NULL;  
252 }
```

02.wo_Defects/memory_leak.c:431

Level Medium**Status** Not processed

```
428 {  
429 char *str = "This is a string";  
430 memory_leak_0016_func_001(strlen(str));  
  
431 strcpy(memory_leak_0016_gbl_ptr,str);  
  
432 free(memory_leak_0016_gbl_ptr);  
433 memory_leak_0016_gbl_ptr = NULL;  
434 }
```

Trace

vflag == 1

02.wo_Defects/memory_leak.c:548

```
545 extern volatile int vflag;  
546 void memory_leak_main ()  
547 {  
  
548 if (vflag == 1 || vflag ==888)  
  
549 {  
550     memory_leak_001();  
551 }
```

```
strcpy(memory_leak_0016_gbl_ptr,str)
```

02.wo_Defects/memory_leak.c:431

```
428 {  
429   char *str = "This is a string";  
430   memory_leak_0016_func_001(strlen(str));  
  
431   strcpy(memory_leak_0016_gbl_ptr,str);  
  
432   free(memory_leak_0016_gbl_ptr);  
433   memory_leak_0016_gbl_ptr = NULL;  
434 }
```

02.wo_Defects/null_pointer.c:259

Level Medium

Status Not processed

```
256 {  
257   char *str = "This is a string";  
258   null_pointer_015_func_001(strlen(str));  
  
259   strcpy(null_pointer_015_gbl_ptr,str); /*Tool should not detect this line as error*/  
/*NO ERROR:NULL pointer dereference*/  
  
260   free(null_pointer_015_gbl_ptr);  
261   null_pointer_015_gbl_ptr = NULL;  
262 }
```

Trace

```
vflag == 1
```

02.wo_Defects/null_pointer.c:375

```
372 extern volatile int vflag;  
373 void null_pointer_main ()  
374 {
```

```
375 if (vflag == 1 || vflag ==888)
```

```
376 {  
377     null_pointer_001();  
378 }
```

```
strcpy(null_pointer_015_gbl_ptr,str)
```

02.wo_Defects/null_pointer.c:259

```
256 {  
257     char *str = "This is a string";  
258     null_pointer_015_func_001(strlen(str));
```

```
259 strcpy(null_pointer_015_gbl_ptr,str); /*Tool should not detect this line as  
error*/ /*NO ERROR:NULL pointer dereference*/
```

```
260 free(null_pointer_015_gbl_ptr);  
261 null_pointer_015_gbl_ptr = NULL;  
262 }
```

Identical expressions (C/C++)

Description

One of the following code constructions encountered:

- Identical expressions are on both sides of bitwise or logical operator;
- True and false branches are identical;
- Identical expressions are compared;
- Identical expressions on both sides of ':' conditional expression;
- Condition is unnecessarily checked twice.

This may indicate an error in program logic.

Example

In the following example, the code that has violated the execution logic:

```
void f(int x, int y) {  
    x = x | x;  
    x = y?x:x;  
    if(x)  
        return;
```

```
    else
        return;
    if(x == 1) {
        if(x == 1)
            y = 3;
    }
}
```

Recommendations

- Make sure that there is no violation of the code execution logic.

Links

1. Probable logical error

Vulnerability Entries

01.w_Defects/pow_related_errors.c:441

Level Medium

Status Not processed

```
438
439 void pow_related_errors_026()
440 {
```

```
441 double base=10^10;
```

```
442 double exponent=7000;
443
444 double ans;
```

Trace

```
double base=10^10
```

```
01.w_Defects/pow_related_errors.c:441
```

```
438
439 void pow_related_errors_026()
440 {

441 double base=10^10;

442 double exponent=7000;
443
444 double ans;
```

```
double base=10^10
```

```
01.w_Defects/pow_related_errors.c:441
```

```
438
439 void pow_related_errors_026()
440 {

441 double base=10^10;

442 double exponent=7000;
443
444 double ans;
```

Incompatible allocated type (C/C++)

Description

Inconsistencies between the casted type of the return value of a `malloc()`/`calloc()`/`realloc()` call and the operand of `sizeof()` expression contained within its argument can lead to errors.

Example

In the following example, the size of data type A and B are different. Therefore, an attempt to use a pointer will result in an error:

```
#define N 100000
```

```
struct A{int data[N];};
```

```
struct B{};  
  
int main()  
{  
    struct A *a = calloc(1, sizeof(struct B));  
    for(int i = 0; i < N; ++i)  
        a->data[i] = i;  
}
```

Recommendations

- Keep track of which pointers to which types of structures you use.
- Use proven libraries that do not allow this vulnerability.
- Use tools that allow you to dynamically track memory management problems (for example, valgrind).

Links

1. Dynamic Memory Allocation in C
2. C Programming Language: Functions?—?malloc(), calloc(), realloc() and free().

Vulnerability Entries

01.w_Defects/uninit_pointer.c:215

Level Medium

Status Not processed

```
212 void uninit_pointer_011 ()  
213 {  
214     unsigned int * ptr,a=0;  
  
215     ptr = (unsigned int *)malloc(10*sizeof(unsigned int ));  
  
216     int i;  
217     if (ptr!=NULL)  
218     {
```

Trace

```
sizeof(unsigned int *) unsigned int *
```

01.w_Defects/uninit_pointer.c:215

```
212 void uninit_pointer_011 ()  
213 {  
214     unsigned int * ptr,a=0;  
  
215     ptr = (unsigned int *)malloc(10*sizeof(unsigned int));  
  
216     int i;  
217     if (ptr!=NULL)  
218     {
```

```
sizeof(unsigned int *) unsigned int *
```

01.w_Defects/uninit_pointer.c:215

```
212 void uninit_pointer_011 ()  
213 {  
214     unsigned int * ptr,a=0;  
  
215     ptr = (unsigned int *)malloc(10*sizeof(unsigned int));  
  
216     int i;  
217     if (ptr!=NULL)  
218     {
```

02.wo_Defects/uninit_pointer.c:227

Level Medium

Status Not processed

```
224 void uninit_pointer_011 ()  
225 {  
226     unsigned int * ptr = NULL,a=0;  
  
227     ptr = (unsigned int *)malloc(10*sizeof(unsigned int));  
  
228     int i;
```

```
229 if (ptr!=NULL)
230 {
```

Trace

```
sizeof(unsigned int *) unsigned int *
```

02.wo_Defects/uninit_pointer.c:227

```
224 void uninit_pointer_011 ()
225 {
226     unsigned int * ptr = NULL,a=0;

227     ptr = (unsigned int *)malloc(10*sizeof(unsigned int));

228     int i;
229     if (ptr!=NULL)
230 {
```

```
sizeof(unsigned int *) unsigned int *
```

02.wo_Defects/uninit_pointer.c:227

```
224 void uninit_pointer_011 ()
225 {
226     unsigned int * ptr = NULL,a=0;

227     ptr = (unsigned int *)malloc(10*sizeof(unsigned int));

228     int i;
229     if (ptr!=NULL)
230 {
```

Insecure string API (C/C++)

Description

The string manipulation function used is insecure, since it allows a buffer overflow. Insecure functions include: strcpy, strcat. This may lead to incorrect behavior of the application, crash, or violation of valuable data confidentiality.

The `strcpy()` function copies the C string pointed by `source` into the array pointed by `destination`, including the terminating null character. The `strcat` appends a copy of the source string to the destination string. It is important to note that, the destination array should be large enough otherwise it may result in undefined behavior.

Example

In the following example, the application uses an insecure string buffer copy function:

```
char x[4];
char *y;
strcpy(x, y);
```

Recommendations

- Use secure analogues (`strncpy`).

Links

1. CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer
2. `strcpy` — cplusplus.com
3. `strcat` — cplusplus.com

Vulnerability Entries

01.w_Defects/free_null_pointer.c:109

Level Medium

Status Not processed

```
106 {
107     char *str = "This is a string";
108     free_null_pointer_005_func_001(strlen(str));

109     strcpy(free_null_pointer_005_gbl_ptr,str);

110     free(free_null_pointer_005_gbl_ptr); /* Tool should detect this line as error
*/*ERROR:Freeing a NULL pointer*/
111     free_null_pointer_005_gbl_ptr = NULL;
112 }
```

Trace

```
strcpy(free_null_pointer_005_gbl_ptr,str)
```

01.w_Defects/free_null_pointer.c:109

```
106 {  
107     char *str = "This is a string";  
108     free_null_pointer_005_func_001(strlen(str));  
  
109     strcpy(free_null_pointer_005_gbl_ptr,str);  
  
110     free(free_null_pointer_005_gbl_ptr); /* Tool should detect this line as error  
*////*ERROR:Freeing a NULL pointer*/  
111     free_null_pointer_005_gbl_ptr = NULL;  
112 }
```

```
strcpy(free_null_pointer_005_gbl_ptr,str)
```

01.w_Defects/free_null_pointer.c:109

```
106 {  
107     char *str = "This is a string";  
108     free_null_pointer_005_func_001(strlen(str));  
  
109     strcpy(free_null_pointer_005_gbl_ptr,str);  
  
110     free(free_null_pointer_005_gbl_ptr); /* Tool should detect this line as error  
*////*ERROR:Freeing a NULL pointer*/  
111     free_null_pointer_005_gbl_ptr = NULL;  
112 }
```

01.w_Defects/free_null_pointer.c:146

Level Medium

Status Not processed

```
143     {  
144         (s+i)->buf = NULL;  
145     }  
  
146     strcpy((s+4)->buf,s1);
```

```
147 }  
148 if(free_null_pointer_006_func_001(flag)==0)  
149 {
```

Trace

```
strcpy((s+4)->buf,s1)
```

01.w_Defects/free_null_pointer.c:146

```
143 {  
144     (s+i)->buf = NULL;  
145 }
```

```
146 strcpy((s+4)->buf,s1);
```

```
147 }  
148 if(free_null_pointer_006_func_001(flag)==0)  
149 {
```

```
strcpy((s+4)->buf,s1)
```

01.w_Defects/free_null_pointer.c:146

```
143 {  
144     (s+i)->buf = NULL;  
145 }
```

```
146 strcpy((s+4)->buf,s1);
```

```
147 }  
148 if(free_null_pointer_006_func_001(flag)==0)  
149 {
```

01.w_Defects/free_null_pointer.c:241

Level Medium

Status Not processed

```
238 char *str = "This is a string";
239 char *str1=NULL;
240 free_null_pointer_008_func_001(strlen(str),&str1);

241 strcpy(str1,str);

242 free(str1);/* Tool should detect this line as error *//*ERROR:Freeing a NULL
pointer*/
243 str1 = NULL;
244 }
```

Trace

```
strcpy(str1,str)
```

01.w_Defects/free_null_pointer.c:241

```
238 char *str = "This is a string";
239 char *str1=NULL;
240 free_null_pointer_008_func_001(strlen(str),&str1);

241 strcpy(str1,str);
```

```
242 free(str1);/* Tool should detect this line as error *//*ERROR:Freeing a NULL
pointer*/
243 str1 = NULL;
244 }
```

```
strcpy(str1,str)
```

01.w_Defects/free_null_pointer.c:241

```
238 char *str = "This is a string";
239 char *str1=NULL;
240 free_null_pointer_008_func_001(strlen(str),&str1);

241 strcpy(str1,str);
```

```
242 free(str1);/* Tool should detect this line as error *//*ERROR:Freeing a NULL
pointer*/
243 str1 = NULL;
244 }
```

01.w_Defects/free_null_pointer.c:275

Level Medium**Status** Not processed

```
272 free_null_pointer_009_func_001();
273 for(i=0;i<5;i++)
274 {
275     strcpy (free_null_pointer_009dst[i],"STRING");
276 }
277 while(1)
278 {
```

Trace

```
strcpy (free_null_pointer_009dst[i],"  
STRING")
```

01.w_Defects/free_null_pointer.c:275

```
272 free_null_pointer_009_func_001();
273 for(i=0;i<5;i++)
274 {
275     strcpy (free_null_pointer_009dst[i],"STRING");
276 }
277 while(1)
278 {
```

```
strcpy (free_null_pointer_009dst[i],"  
STRING")
```

01.w_Defects/free_null_pointer.c:275

```
272 free_null_pointer_009_func_001();
273 for(i=0;i<5;i++)
274 {
275     strcpy (free_null_pointer_009dst[i],"STRING");
```

```
276 }
277 while(1)
278 {
```

01.w_Defects/free_null_pointer.c:320

Level Medium

Status Not processed

```
317 int i;
318 for(i=0;i<5;i++)
319 {
320     strcpy(*(dst+i),src[i]);
321 }
322 }
323 void free_null_pointer_010_func_003()
```

Trace

```
strcpy(*(dst+i),src[i])
```

01.w_Defects/free_null_pointer.c:320

```
317 int i;
318 for(i=0;i<5;i++)
319 {
320     strcpy(*(dst+i),src[i]);
321 }
322 }
323 void free_null_pointer_010_func_003()
```

```
strcpy(*(dst+i),src[i])
```

01.w_Defects/free_null_pointer.c:320

```
317 int i;
318 for(i=0;i<5;i++)
319 {
320     strcpy(*(dst+i),src[i]);
321 }
322 }
323 void free_null_pointer_010_func_003()
```

01.w_Defects/invalid_memory_access.c:102

Level Medium

Status Not processed

```
99 buf = (char *) malloc (25 * sizeof(char));
100 if(buf!=NULL)
101 {
102     strcpy(buf,"This is String");
103     free(buf);
104 }
105 c = buf; /*Tool should detect this line as error*/ /*ERROR:Invalid memory access to
already freed area*/
```

Trace

```
strcpy(buf,"This is String")
```

01.w_Defects/invalid_memory_access.c:102

```
99 buf = (char *) malloc (25 * sizeof(char));
100 if(buf!=NULL)
101 {
```

```
102 strcpy(buf,"This is String");

103 free(buf);
104 }
105 c = buf; /*Tool should detect this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

```
strcpy(buf,"This is String")
```

01.w_Defects/invalid_memory_access.c:102

```
99 buf = (char *) malloc (25 * sizeof(char));
100 if(buf!=NULL)
101 {

102 strcpy(buf,"This is String");

103 free(buf);
104 }
105 c = buf; /*Tool should detect this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

01.w_Defects/invalid_memory_access.c:205

Level Medium

Status Not processed

```
202
203 for(i=0;i<5;i++)
204 {

205 strcpy(*(ptr+i),"String");
```

```
206 free(ptr[i]);
207 ptr[i] = NULL;
208 }
```

Trace

```
strcpy(*(ptr+i),"String")
```

01.w_Defects/invalid_memory_access.c:205

```
202  
203 for(i=0;i<5;i++)  
204 {  
  
205 strcpy(*(ptr+i),"String");
```

```
206     free(ptr[i]);  
207     ptr[i] = NULL;  
208 }
```

```
strcpy(*(ptr+i),"String")
```

01.w_Defects/invalid_memory_access.c:205

```
202  
203 for(i=0;i<5;i++)  
204 {  
  
205 strcpy(*(ptr+i),"String");
```

```
206     free(ptr[i]);  
207     ptr[i] = NULL;  
208 }
```

01.w_Defects/invalid_memory_access.c:210

Level Medium

Status Not processed

```
207     ptr[i] = NULL;  
208 }  
209 free(ptr);
```

```
210 strcpy(*(ptr+2),"String"); /*Tool should detect this line as error*/ /*ERROR:Invalid  
memory access to already freed area*/
```

```
211 }  
212  
213 /*
```

Trace

```
strcpy(*(ptr+2),"String")
```

01.w_Defects/invalid_memory_access.c:210

```
207     ptr[i] = NULL;  
208 }  
209 free(ptr);
```

210 strcpy(*(ptr+2),"String"); /*Tool should detect this line as error*/ /*ERROR:
Invalid memory access to already freed area*/

```
211 }  
212  
213 /*
```

```
strcpy(*(ptr+2),"String")
```

01.w_Defects/invalid_memory_access.c:210

```
207     ptr[i] = NULL;  
208 }  
209 free(ptr);
```

210 strcpy(*(ptr+2),"String"); /*Tool should detect this line as error*/ /*ERROR:
Invalid memory access to already freed area*/

```
211 }  
212  
213 /*
```

01.w_Defects/invalid_memory_access.c:560

Level Medium

Status Not processed

```
557         for(i=0;i<10;i++)  
558     {  
559             invalid_memory_access_016_doubleptr_gbl[i]=(char*) malloc  
(10*sizeof(char));  
  
560             strcpy(invalid_memory_access_016_doubleptr_gbl[i],"  
STRING00");  
  
561     }  
562 }  
563 }
```

Trace

```
strcpy  
(invalid_memory_access_016_doubleptr_gb  
l[i],"STRING00")
```

01.w_Defects/invalid_memory_access.c:560

```
557         for(i=0;i<10;i++)  
558     {  
559             invalid_memory_access_016_doubleptr_gbl[i]=(char*) malloc  
(10*sizeof(char));  
  
560             strcpy(invalid_memory_access_016_doubleptr_gbl[i],"  
STRING00");  
  
561     }  
562 }  
563 }
```

```
strcpy  
(invalid_memory_access_016_doubleptr_gb  
l[i],"STRING00")
```

01.w_Defects/invalid_memory_access.c:560

```
557         for(i=0;i<10;i++)  
558     {  
559             invalid_memory_access_016_doubleptr_gbl[i]=(char*) malloc  
(10*sizeof(char));  
  
560             strcpy(invalid_memory_access_016_doubleptr_gbl[i],"
```

```
STRING00");
```

```
561      }
562  }
563 }
```

01.w_Defects/invalid_memory_access.c:568

Level Medium

Status Not processed

```
565 void invalid_memory_access_016_func_003()
566 {
567   char s[10] ;
```

568 strcpy(s,invalid_memory_access_016_doubleptr_gbl[0]);/*Tool should detect this
line as error*/ /*ERROR:Invalid memory access to already freed area*/

```
569 }
570
571 void invalid_memory_access_016()
```

Trace

```
strcpy(s,
invalid_memory_access_016_doubleptr_gbl
[0])
```

01.w_Defects/invalid_memory_access.c:568

```
565 void invalid_memory_access_016_func_003()
566 {
567   char s[10] ;
```

568 strcpy(s,invalid_memory_access_016_doubleptr_gbl[0]);/*Tool should detect
this line as error*/ /*ERROR:Invalid memory access to already freed area*/

```
569 }
570
571 void invalid_memory_access_016()
```

```
strcpy(s,  
invalid_memory_access_016_doubleptr_gbl  
[0])
```

01.w_Defects/invalid_memory_access.c:568

```
565 void invalid_memory_access_016_func_003()  
566 {  
567   char s[10] ;
```

```
568   strcpy(s,invalid_memory_access_016_doubleptr_gbl[0]);/*Tool should detect  
this line as error*/ /*ERROR:Invalid memory access to already freed area*/
```

```
569 }  
570  
571 void invalid_memory_access_016()
```

01.w_Defects/invalid_memory_access.c:611

Level Medium

Status Not processed

```
608 {  
609   invalid_memory_access_017_doubleptr_gbl=(char*) malloc(10*sizeof(char));  
610   if(invalid_memory_access_017_doubleptr_gbl !=NULL)
```

```
611   strcpy(invalid_memory_access_017_doubleptr_gbl,"TEST");
```

```
612 }  
613  
614 void invalid_memory_access_017_func_003()
```

Trace

```
strcpy
(invalid_memory_access_017_doubleptr_gb
I,"TEST")
```

01.w_Defects/invalid_memory_access.c:611

```
608 {
609 invalid_memory_access_017_doubleptr_gbl=(char*) malloc(10*sizeof(char));
610 if(invalid_memory_access_017_doubleptr_gbl !=NULL)

611 strcpy(invalid_memory_access_017_doubleptr_gbl,"TEST");

612 }
613
614 void invalid_memory_access_017_func_003()
```

```
strcpy
(invalid_memory_access_017_doubleptr_gb
I,"TEST")
```

01.w_Defects/invalid_memory_access.c:611

```
608 {
609 invalid_memory_access_017_doubleptr_gbl=(char*) malloc(10*sizeof(char));
610 if(invalid_memory_access_017_doubleptr_gbl !=NULL)

611 strcpy(invalid_memory_access_017_doubleptr_gbl,"TEST");

612 }
613
614 void invalid_memory_access_017_func_003()
```

01.w_Defects/invalid_memory_access.c:622

Level Medium

Status Not processed

```
619 void invalid_memory_access_017_func_004()
620 {
621 char s[10] ;
```

```
622 strcpy(s,invalid_memory_access_017_doubleptr_gbl);/*Tool should detect this line  
as error*/ /*ERROR:Invalid memory access to already freed area*/
```

```
623 }  
624  
625 void invalid_memory_access_017()
```

Trace

```
strcpy(s,  
invalid_memory_access_017_doubleptr_gbl  
)
```

01.w_Defects/invalid_memory_access.c:622

```
619 void invalid_memory_access_017_func_004()  
620 {  
621   char s[10] ;
```

```
622 strcpy(s,invalid_memory_access_017_doubleptr_gbl);/*Tool should detect  
this line as error*/ /*ERROR:Invalid memory access to already freed area*/
```

```
623 }  
624  
625 void invalid_memory_access_017()
```

```
strcpy(s,  
invalid_memory_access_017_doubleptr_gbl  
)
```

01.w_Defects/invalid_memory_access.c:622

```
619 void invalid_memory_access_017_func_004()  
620 {  
621   char s[10] ;
```

```
622 strcpy(s,invalid_memory_access_017_doubleptr_gbl);/*Tool should detect  
this line as error*/ /*ERROR:Invalid memory access to already freed area*/
```

```
623 }  
624  
625 void invalid_memory_access_017()
```

01.w_Defects/memory_allocation_failure.c:495

Level Medium**Status** Not processed

```
492 if (staticflag == 10)
493     (*(ptr+1) = 10.5);
494 else

495     strcpy( dptr[1],"STRING TEST" );

496
497 if(staticflag == 10)
498     b = *(ptr+1);
```

Trace

strcpy(dptr[1],"STRING TEST")

01.w_Defects/memory_allocation_failure.c:495

```
492 if (staticflag == 10)
493     (*(ptr+1) = 10.5);
494 else

495     strcpy( dptr[1],"STRING TEST" );

496
497 if(staticflag == 10)
498     b = *(ptr+1);
```

strcpy(dptr[1],"STRING TEST")

01.w_Defects/memory_allocation_failure.c:495

```
492 if (staticflag == 10)
493     (*(ptr+1) = 10.5);
494 else

495     strcpy( dptr[1],"STRING TEST" );

496
```

```
497 if(staticflag == 10)
498   b = *(ptr+1);
```

01.w_Defects/memory_allocation_failure.c:577

Level Medium

Status Not processed

```
574 }
575 else
576 {

577   strcpy( dptr[1],"STRING TEST" );

578   if(1)
579   {
580     for (i = 0;i< 4; i++)
```

Trace

```
strcpy( dptr[1],"STRING TEST" )
```

01.w_Defects/memory_allocation_failure.c:577

```
574 }
575 else
576 {

577   strcpy( dptr[1],"STRING TEST" );

578   if(1)
579   {
580     for (i = 0;i< 4; i++)
```

```
strcpy( dptr[1],"STRING TEST" )
```

01.w_Defects/memory_allocation_failure.c:577

```
574 }  
575 else  
576 {  
  
577     strcpy( dptr[1],"STRING TEST" );  
  
578     if(1)  
579     {  
580         for (i = 0;i< 4; i++)
```

01.w_Defects/memory_leak.c:73

Level Medium

Status Not processed

```
70     char *str = "This is a string";  
71     char *str1;  
72     memory_leak_003_func_001(strlen(str),&str1);/*Tool should detect this line as  
error*/ /*ERROR:Memory Leakage */  
  
73     strcpy(str1,str);  
  
74 }  
75  
76 /*
```

Trace

```
strcpy(str1,str)
```

01.w_Defects/memory_leak.c:73

```
70     char *str = "This is a string";  
71     char *str1;  
72     memory_leak_003_func_001(strlen(str),&str1);/*Tool should detect this line  
as error*/ /*ERROR:Memory Leakage */
```

```
73    strcpy(str1,str);
```

```
74 }
```

```
75
```

```
76 /*
```

```
strcpy(str1,str)
```

01.w_Defects/memory_leak.c:73

```
70    char *str = "This is a string";
```

```
71    char *str1;
```

```
72    memory_leak_003_func_001(strlen(str),&str1);/*Tool should detect this line  
as error*/ /*ERROR:Memory Leakage */
```

```
73    strcpy(str1,str);
```

```
74 }
```

```
75
```

```
76 /*
```

01.w_Defects/memory_leak.c:96

Level Medium

Status Not processed

```
93    {
```

```
94        (s+i)->buf = (char*)malloc(25* sizeof(char));/*Tool should detect this line as  
error*/ /*ERROR:Memory Leakage */
```

```
95    }
```

```
96    strcpy((s+4)->buf,s1);
```

```
97    for(i= 0; i<5 ;i++);
```

```
98    free(s);
```

```
99 }
```

Trace

```
strcpy((s+4)->buf,s1)
```

01.w_Defects/memory_leak.c:96

```
93  {
94      (s+i)->buf = (char*)malloc(25* sizeof(char));/*Tool should detect this
line as error*/ /*ERROR:Memory Leakage */
95  }

96  strcpy((s+4)->buf,s1);

97  for(i= 0; i<5 ;i++);
98  free(s);
99 }
```

```
strcpy((s+4)->buf,s1)
```

01.w_Defects/memory_leak.c:96

```
93  {
94      (s+i)->buf = (char*)malloc(25* sizeof(char));/*Tool should detect this
line as error*/ /*ERROR:Memory Leakage */
95  }

96  strcpy((s+4)->buf,s1);

97  for(i= 0; i<5 ;i++);
98  free(s);
99 }
```

01.w_Defects/memory_leak.c:270

Level Medium

Status Not processed

```
267  buf = (char *)calloc(50, sizeof(char)); /*Tool should detect this line as error*/
/*ERROR:Memory Leakage */
268 if(buf!=NULL)
269 {
```

```
270  strcpy(buf, "This Is A String");
```

```
271 un.u1 = buf;  
272 }  
273 {
```

Trace

```
strcpy(buf, "This Is A String")
```

01.w_Defects/memory_leak.c:270

```
267 buf = (char *)calloc(50, sizeof(char)); /*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
268 if(buf!=NULL)  
269 {
```

```
270 strcpy(buf, "This Is A String");
```

```
271 un.u1 = buf;  
272 }  
273 {
```

```
strcpy(buf, "This Is A String")
```

01.w_Defects/memory_leak.c:270

```
267 buf = (char *)calloc(50, sizeof(char)); /*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
268 if(buf!=NULL)  
269 {
```

```
270 strcpy(buf, "This Is A String");
```

```
271 un.u1 = buf;  
272 }  
273 {
```

```
01.w_Defects/memory_leak.c:402
```

Level Medium

Status Not processed

```
399 char *str1 = memory_leak_0015_func_001(strlen(str)); /*Tool should detect this line  
as error*/ /*ERROR:Memory Leakage */  
400 if(str1!=NULL)  
401 {  
  
402     strcpy(str1,str);  
  
403 }  
404 }  
405
```

Trace

strcpy(str1,str)

01.w_Defects/memory_leak.c:402

```
399 char *str1 = memory_leak_0015_func_001(strlen(str)); /*Tool should detect  
this line as error*/ /*ERROR:Memory Leakage */  
400 if(str1!=NULL)  
401 {  
  
402     strcpy(str1,str);  
  
403 }  
404 }  
405
```

strcpy(str1,str)

01.w_Defects/memory_leak.c:402

```
399 char *str1 = memory_leak_0015_func_001(strlen(str)); /*Tool should detect  
this line as error*/ /*ERROR:Memory Leakage */  
400 if(str1!=NULL)  
401 {  
  
402     strcpy(str1,str);
```

```
403 }
404 }
405
```

01.w_Defects/memory_leak.c:424

Level Medium

Status Not processed

```
421 {
422 char *str = "This is a string";
423 memory_leak_0016_func_001(strlen(str));

424 strcpy(memory_leak_0016_gbl_ptr,str);

425 }
426
427 /*
```

Trace

```
strcpy(memory_leak_0016_gbl_ptr,str)
```

01.w_Defects/memory_leak.c:424

```
421 {
422 char *str = "This is a string";
423 memory_leak_0016_func_001(strlen(str));

424 strcpy(memory_leak_0016_gbl_ptr,str);

425 }
426
427 /*
```

```
strcpy(memory_leak_0016_gbl_ptr,str)
```

01.w_Defects/memory_leak.c:424

```
421 {  
422     char *str = "This is a string";  
423     memory_leak_0016_func_001(strlen(str));  
  
424     strcpy(memory_leak_0016_gbl_ptr,str);  
  
425 }  
426  
427 /*
```

01.w_Defects/memory_leak.c:517

Level Medium

Status Not processed

```
514     memory_leak_0018_func_001();  
515     for(i=0;i<5;i++)  
516     {  
  
517         strcpy (memory_leak_0018dst[i],"STRING");  
  
518     }  
519 while(1)  
520 {
```

Trace

```
strcpy (memory_leak_0018dst[i],"STRING")
```

01.w_Defects/memory_leak.c:517

```
514     memory_leak_0018_func_001();  
515     for(i=0;i<5;i++)  
516     {  
  
517         strcpy (memory_leak_0018dst[i],"STRING");
```

```
518 }
519 while(1)
520 {
```

strcpy (memory_leak_0018dst[i],"STRING")

01.w_Defects/memory_leak.c:517

```
514 memory_leak_0018_func_001();
515 for(i=0;i<5;i++)
516 {
517     strcpy (memory_leak_0018dst[i],"STRING");
518 }
519 while(1)
520 {
```

01.w_Defects/null_pointer.c:238

Level Medium

Status Not processed

```
235 {
236 char *str = "This is a string";
237 null_pointer_015_func_001(strlen(str));
```

```
238 strcpy(null_pointer_015_gbl_ptr,str);/*Tool should detect this line as error*/
/*ERROR:NULL pointer dereference*/
```

```
239 free(null_pointer_015_gbl_ptr);
240 null_pointer_015_gbl_ptr = NULL;
241 }
```

Trace

```
strcpy(null_pointer_015_gbl_ptr,str)
```

01.w_Defects/null_pointer.c:238

```
235 {  
236   char *str = "This is a string";  
237   null_pointer_015_func_001(strlen(str));  
  
238   strcpy(null_pointer_015_gbl_ptr,str);/*Tool should detect this line as error*/  
/*ERROR:NULL pointer dereference*/  
  
239   free(null_pointer_015_gbl_ptr);  
240   null_pointer_015_gbl_ptr = NULL;  
241 }
```

```
strcpy(null_pointer_015_gbl_ptr,str)
```

01.w_Defects/null_pointer.c:238

```
235 {  
236   char *str = "This is a string";  
237   null_pointer_015_func_001(strlen(str));  
  
238   strcpy(null_pointer_015_gbl_ptr,str);/*Tool should detect this line as error*/  
/*ERROR:NULL pointer dereference*/  
  
239   free(null_pointer_015_gbl_ptr);  
240   null_pointer_015_gbl_ptr = NULL;  
241 }
```

01.w_Defects/null_pointer.c:334

Level Medium

Status Not processed

```
331   null_pointer_017_func_001(0);  
332   for(i=0;i<5;i++)  
333   {
```

```
334       strcpy (null_pointer_017dst[i],"STRING");/*Tool should detect this line as  
error*/ /*ERROR:NULL pointer dereference*/
```

```
335 }
336 while(1)
337 {
```

Trace

```
strcpy (null_pointer_017dst[i],"STRING")
```

01.w_Defects/null_pointer.c:334

```
331 null_pointer_017_func_001(0);
332 for(i=0;i<5;i++)
333 {
```

```
334     strcpy (null_pointer_017dst[i],"STRING");/*Tool should detect this
line as error*/ /*ERROR:NULL pointer dereference*/
```

```
335 }
336 while(1)
337 {
```

```
strcpy (null_pointer_017dst[i],"STRING")
```

01.w_Defects/null_pointer.c:334

```
331 null_pointer_017_func_001(0);
332 for(i=0;i<5;i++)
333 {
```

```
334     strcpy (null_pointer_017dst[i],"STRING");/*Tool should detect this
line as error*/ /*ERROR:NULL pointer dereference*/
```

```
335 }
336 while(1)
337 {
```

```
01.w_Defects/st_cross_thread_access.c:335
```

Level Medium

Status Not processed

```
332 if (ip >= 0)
333 {
334 pthread_mutex_lock( &st_cross_thread_access_004_glb_mutex_2 );

335 strcpy(*pbuff[0],"TEST");/*Tool should detect this line as error*/ /*ERROR:Cross
thread stack access error*/

336 pthread_mutex_unlock( &st_cross_thread_access_004_glb_mutex_2 );
337 }
338 i--;
```

Trace

strcpy(*pbuff[0],"TEST")

01.w_Defects/st_cross_thread_access.c:335

```
332 if (ip >= 0)
333 {
334 pthread_mutex_lock( &st_cross_thread_access_004_glb_mutex_2 );

335 strcpy(*pbuff[0],"TEST");/*Tool should detect this line as error*/ /*ERROR:
Cross thread stack access error*/

336 pthread_mutex_unlock( &st_cross_thread_access_004_glb_mutex_2 );
337 }
338 i--;
```

strcpy(*pbuff[0],"TEST")

01.w_Defects/st_cross_thread_access.c:335

```
332 if (ip >= 0)
333 {
334 pthread_mutex_lock( &st_cross_thread_access_004_glb_mutex_2 );

335 strcpy(*pbuff[0],"TEST");/*Tool should detect this line as error*/ /*ERROR:
Cross thread stack access error*/
```

```
336 pthread_mutex_unlock( &st_cross_thread_access_004_glb_mutex_2 );
337 }
338 i--;
```

01.w_Defects/st_underrun.c:80

Level Medium

Status Not processed

```
77 void st_underrun_003_func_001 (st_underrun_003_s_001 *s)
78 {
79     char buf[10] = "STRING";
80     strcpy(s->buf,buf);
81 }
82
83 void st_underrun_003_func_002 (st_underrun_003_s_001 *s)
```

Trace

strcpy(s->buf,buf)

01.w_Defects/st_underrun.c:80

```
77 void st_underrun_003_func_001 (st_underrun_003_s_001 *s)
78 {
79     char buf[10] = "STRING";
80     strcpy(s->buf,buf);
81 }
82
83 void st_underrun_003_func_002 (st_underrun_003_s_001 *s)
```

```
strcpy(s->buf,buf)
```

01.w_Defects/st_underrun.c:80

```
77 void st_underrun_003_func_001 (st_underrun_003_s_001 *s)
78 {
79     char buf[10] = "STRING";
80     strcpy(s->buf,buf);
81 }
82
83 void st_underrun_003_func_002 (st_underrun_003_s_001 *s)
```

01.w_Defects/st_underrun.c:115

Level Medium

Status Not processed

```
112 void st_underrun_004_func_002 (st_underrun_004_s_001 *s)
113 {
114     char buf[10] = "STRING";
115     strcpy(s->buf,buf);
116 }
117
118 st_underrun_004_s_001 st_underrun_004_func_001 (st_underrun_004_s_001 *s)
```

Trace

```
strcpy(s->buf,buf)
```

01.w_Defects/st_underrun.c:115

```
112 void st_underrun_004_func_002 (st_underrun_004_s_001 *s)
113 {
114     char buf[10] = "STRING";
115     strcpy(s->buf,buf);
```

```
116 }
117
118 st_underrun_004_s_001 st_underrun_004_func_001
(st_underrun_004_s_001 *s)
```

```
strcpy(s->buf,buf)
```

01.w_Defects/st_underrun.c:115

```
112 void st_underrun_004_func_002 (st_underrun_004_s_001 *s)
113 {
114     char buf[10] = "STRING";

115     strcpy(s->buf,buf);

116 }
117
118 st_underrun_004_s_001 st_underrun_004_func_001
(st_underrun_004_s_001 *s)
```

01.w_Defects/st_underrun.c:204

Level Medium

Status Not processed

```
201 void st_underrun_006 ()
202 {
203     st_underrun_006_s_001 s;

204     strcpy(s.buf,"STRING !!!!");

205     void (*func)(st_underrun_006_s_001);
206     func = st_underrun_006_func_001;
207     func(s);
```

Trace

```
strcpy(s.buf,"STRING !!!!")
```

01.w_Defects/st_underrun.c:204

```
201 void st_underrun_006 ()  
202 {  
203   st_underrun_006_s_001 s;  
  
204   strcpy(s.buf,"STRING !!!");  
  
205   void (*func)(st_underrun_006_s_001);  
206   func = st_underrun_006_func_001;  
207   func(s);
```

```
strcpy(s.buf,"STRING !!!!")
```

01.w_Defects/st_underrun.c:204

```
201 void st_underrun_006 ()  
202 {  
203   st_underrun_006_s_001 s;  
  
204   strcpy(s.buf,"STRING !!!");  
  
205   void (*func)(st_underrun_006_s_001);  
206   func = st_underrun_006_func_001;  
207   func(s);
```

01.w_Defects/uninit_memory_access.c:54

Level Medium

Status Not processed

```
51 char *str2 ;  
52 if (str1!=NULL)  
53 {  
  
54   strcpy(str1, str2);  
  
55   printf("%s %s\n",str1,str2);/*Tool should detect this line as error*/ /*ERROR:
```

Uninitialized Memory Access*/

```
56     free(str1);
57 }
```

Trace

strcpy(str1, str2)

01.w_Defects/uninit_memory_access.c:54

```
51 char *str2 ;
52 if (str1!=NULL)
53 {

54     strcpy(str1, str2);

55     printf("%s %s\n",str1,str2);/*Tool should detect this line as error*/
/*ERROR:Uninitialized Memory Access*/
56     free(str1);
57 }
```

strcpy(str1, str2)

01.w_Defects/uninit_memory_access.c:54

```
51 char *str2 ;
52 if (str1!=NULL)
53 {

54     strcpy(str1, str2);

55     printf("%s %s\n",str1,str2);/*Tool should detect this line as error*/
/*ERROR:Uninitialized Memory Access*/
56     free(str1);
57 }
```

01.w_Defects/uninit_pointer.c:187

Level Medium

Status Not processed

```
184  }
185  if(uninit_pointer_009_func_001(flag)>0)
186  {

187      strcpy(buf1,buf);/*Tool should detect this line as error*/ /*ERROR:
Uninitialized pointer*/

188  }
189 }
190
```

Trace

strcpy(buf1,buf)

01.w_Defects/uninit_pointer.c:187

```
184  }
185  if(uninit_pointer_009_func_001(flag)>0)
186  {

187      strcpy(buf1,buf);/*Tool should detect this line as error*/ /*ERROR:
Uninitialized pointer*/

188  }
189 }
190
```

strcpy(buf1,buf)

01.w_Defects/uninit_pointer.c:187

```
184  }
185  if(uninit_pointer_009_func_001(flag)>0)
186  {

187      strcpy(buf1,buf);/*Tool should detect this line as error*/ /*ERROR:
Uninitialized pointer*/

188  }
189 }
190
```

01.w_Defects/uninit_pointer.c:385

Level Medium**Status** Not processed

```
382 {  
383     uninit_pointer_016_gbl_doubleptr[i]=(char*) malloc(10*sizeof(char));  
384     if(i<5)  
  
385         strcpy(uninit_pointer_016_gbl_doubleptr[i],"STRING00");  
  
386     }  
387 }  
388 }
```

Trace

```
strcpy(uninit_pointer_016_gbl_doubleptr[i],"  
STRING00")
```

01.w_Defects/uninit_pointer.c:385

```
382 {  
383     uninit_pointer_016_gbl_doubleptr[i]=(char*) malloc(10*sizeof  
(char));  
384     if(i<5)  
  
385         strcpy(uninit_pointer_016_gbl_doubleptr[i],"STRING00");  
  
386     }  
387 }  
388 }
```

```
strcpy(uninit_pointer_016_gbl_doubleptr[i],"  
STRING00")
```

01.w_Defects/uninit_pointer.c:385

```
382 {  
383     uninit_pointer_016_gbl_doubleptr[i]=(char*) malloc(10*sizeof  
(char));  
384     if(i<5)
```

```
385         strcpy(uninit_pointer_016_gbl_doubleptr[i],"STRING00");  
  
386     }  
387 }  
388 }
```

01.w_Defects/uninit_pointer.c:406

Level Medium

Status Not processed

```
403 if(uninit_pointer_016_gbl_doubleptr[i] !=NULL)  
404 {  
405   if(i==7)  
  
406   strcpy(s,uninit_pointer_016_gbl_doubleptr[i]);/*Tool should detect this line as  
error*/ /*ERROR:Uninitialized pointer*/  
  
407 free (uninit_pointer_016_gbl_doubleptr[i]);  
408 }  
409 }
```

Trace

strcpy(s,uninit_pointer_016_gbl_doubleptr[i])

01.w_Defects/uninit_pointer.c:406

```
403 if(uninit_pointer_016_gbl_doubleptr[i] !=NULL)  
404 {  
405   if(i==7)  
  
406   strcpy(s,uninit_pointer_016_gbl_doubleptr[i]);/*Tool should detect this line as  
error*/ /*ERROR:Uninitialized pointer*/  
  
407 free (uninit_pointer_016_gbl_doubleptr[i]);  
408 }  
409 }
```

```
strcpy(s,uninit_pointer_016_gbl_doubleptr[i])
```

01.w_Defects/uninit_pointer.c:406

```
403 if(uninit_pointer_016_gbl_doubleptr[i] !=NULL)
404 {
405   if(i==7)

406   strcpy(s,uninit_pointer_016_gbl_doubleptr[i]);/*Tool should detect this line
as error*/ /*ERROR:Uninitialized pointer*/

407 free (uninit_pointer_016_gbl_doubleptr[i]);
408 }
409 }
```

01.w_Defects/uninit_var.c:142

Level Medium

Status Not processed

```
139 void uninit_var_009_func_001 (char buf[])
140 {
141   char ret[25];

142   strcpy(buf,ret);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
Variable*/

143 }
144
145 void uninit_var_009 ()
```

Trace

```
strcpy(buf,ret)
```

01.w_Defects/uninit_var.c:142

```
139 void uninit_var_009_func_001 (char buf[])
140 {
141   char ret[25];
```

```
142 strcpy(buf,ret);/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/
```

```
143 }
144
145 void uninit_var_009 ()
```

```
strcpy(buf,ret)
```

01.w_Defects/uninit_var.c:142

```
139 void uninit_var_009_func_001 (char buf[])
140 {
141     char ret[25];
```

```
142 strcpy(buf,ret);/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/
```

```
143 }
144
145 void uninit_var_009 ()
```

01.w_Defects/wrong_arguments_func_pointer.c:213

Level Medium

Status Not processed

```
210 */
211 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2, char*str3)
212 {
```

```
213     strcat(str1,str2);
```

```
214     strcpy(str3,str1);
215     return ('c');
216 }
```

Trace

```
strcat(str1,str2)
```

01.w_Defects/wrong_arguments_func_pointer.c:213

```
210 */
211 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2,
char*str3)
212 {

213     strcat(str1,str2);

214     strcpy(str3,str1);
215     return ('c');
216 }
```

```
strcat(str1,str2)
```

01.w_Defects/wrong_arguments_func_pointer.c:213

```
210 */
211 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2,
char*str3)
212 {

213     strcat(str1,str2);

214     strcpy(str3,str1);
215     return ('c');
216 }
```

01.w_Defects/wrong_arguments_func_pointer.c:214

Level Medium

Status Not processed

```
211 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2, char*str3)
212 {
213     strcat(str1,str2);

214     strcpy(str3,str1);
```

```
215 return ('c');
216 }
217 void wrong_arguments_func_pointer_009 ()
```

Trace

```
strcpy(str3,str1)
```

01.w_Defects/wrong_arguments_func_pointer.c:214

```
211 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2,
char*str3)
212 {
213     strcat(str1,str2);

214     strcpy(str3,str1);

215     return ('c');
216 }
217 void wrong_arguments_func_pointer_009 ()
```

```
strcpy(str3,str1)
```

01.w_Defects/wrong_arguments_func_pointer.c:214

```
211 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2,
char*str3)
212 {
213     strcat(str1,str2);

214     strcpy(str3,str1);

215     return ('c');
216 }
217 void wrong_arguments_func_pointer_009 ()
```

01.w_Defects/wrong_arguments_func_pointer.c:382

Level Medium

Status Not processed

```
379 void (*fptr)(char *);  
380 fptr = (void (*)(char*))wrong_arguments_func_pointer_013_func_001;  
381 fptr(str1);/*Tool should detect this line as error**/*ERROR:Wrong arguments passed  
to a function pointer*/  
  
382 strcpy(str1,str);  
  
383 free(str1);  
384 str1 = NULL;  
385
```

Trace

```
strcpy(str1,str)
```

01.w_Defects/wrong_arguments_func_pointer.c:382

```
379 void (*fptr)(char *);  
380 fptr = (void (*)(char*))wrong_arguments_func_pointer_013_func_001;  
381 fptr(str1);/*Tool should detect this line as error**/*ERROR:Wrong arguments passed  
to a function pointer*/
```

```
382 strcpy(str1,str);
```

```
383 free(str1);  
384 str1 = NULL;  
385
```

```
strcpy(str1,str)
```

01.w_Defects/wrong_arguments_func_pointer.c:382

```
379 void (*fptr)(char *);  
380 fptr = (void (*)(char*))wrong_arguments_func_pointer_013_func_001;  
381 fptr(str1);/*Tool should detect this line as error**/*ERROR:Wrong arguments passed  
to a function pointer*/
```

```
382 strcpy(str1,str);
```

```
383 free(str1);  
384 str1 = NULL;  
385
```

01.w_Defects/wrong_arguments_func_pointer.c:439

Level Medium**Status** Not processed

```
436 int i;
437 for(i=0;i<5;i++)
438 {
439     strcpy(*(wrong_arguments_func_pointer_015_dst1_gbl+i),src[i]);
440 }
441 }
442
```

Trace

```
strcpy(*  
(wrong_arguments_func_pointer_015_dst1_  
gbl+i),src[i])
```

01.w_Defects/wrong_arguments_func_pointer.c:439

```
436 int i;
437 for(i=0;i<5;i++)
438 {
```

```
439     strcpy(*(wrong_arguments_func_pointer_015_dst1_gbl+i),src[i]);
440 }
441 }
442
```

```
strcpy(*  
(wrong_arguments_func_pointer_015_dst1_  
gbl+i),src[i])
```

01.w_Defects/wrong_arguments_func_pointer.c:439

```
436 int i;
437 for(i=0;i<5;i++)
438 {
```

```
439     strcpy(*(wrong_arguments_func_pointer_015_dst1_gbl+i),src[i]);  
  
440 }  
441 }  
442
```

01.w_Defects/wrong_arguments_func_pointer.c:478

Level Medium

Status Not processed

```
475 char wrong_arguments_func_pointer_016_func_001(char *str1, int *str2, float*str3)  
476 {  
477     char s[20];  
  
478     strcpy(s,str1);  
  
479     *str2 +=1;  
480     *str3 +=1;  
481     return (*str2);
```

Trace

strcpy(s,str1)

01.w_Defects/wrong_arguments_func_pointer.c:478

```
475 char wrong_arguments_func_pointer_016_func_001(char *str1, int *str2,  
float*str3)  
476 {  
477     char s[20];  
  
478     strcpy(s,str1);  
  
479     *str2 +=1;  
480     *str3 +=1;  
481     return (*str2);
```

```
strcpy(s,str1)
```

01.w_Defects/wrong_arguments_func_pointer.c:478

```
475 char wrong_arguments_func_pointer_016_func_001(char *str1, int *str2,  
float*str3)  
476 {  
477     char s[20];  
  
478     strcpy(s,str1);  
  
479     *str2 +=1;  
480     *str3 +=1;  
481     return (*str2);
```

02.wo_Defects/free_null_pointer.c:26

Level Medium

Status Not processed

```
23 char* buf= (char*) malloc(10*sizeof(char));  
24     if(buf!=NULL)  
25     {  
  
26         strcpy(buf,"STRING");  
  
27         free(buf);/* Tool should not detect this line as error */ /*No ERROR:Freeing a NULL  
pointer*/  
28         buf = NULL;  
29     }
```

Trace

```
strcpy(buf,"STRING")
```

02.wo_Defects/free_null_pointer.c:26

```
23 char* buf= (char*) malloc(10*sizeof(char));  
24     if(buf!=NULL)  
25     {
```

```
26 strcpy(buf,"STRING");

27 free(buf);/* Tool should not detect this line as error */ /*No ERROR:Freeing a
NULL pointer*/
28 buf = NULL;
29 }
```

strcpy(buf,"STRING")

02.wo_Defects/free_null_pointer.c:26

```
23 char* buf= (char*) malloc(10*sizeof(char));
24 if(buf!=NULL)
25 {

26 strcpy(buf,"STRING");

27 free(buf);/* Tool should not detect this line as error */ /*No ERROR:Freeing a
NULL pointer*/
28 buf = NULL;
29 }
```

02.wo_Defects/free_null_pointer.c:119

Level Medium

Status Not processed

```
116 {
117 char *str = "This is a string";
118 free_null_pointer_005_func_001(strlen(str));

119 strcpy(free_null_pointer_005_gbl_ptr,str);

120 free(free_null_pointer_005_gbl_ptr);/* Tool should not detect this line as error */
/*No ERROR:Freeing a NULL pointer*/
121 free_null_pointer_005_gbl_ptr = NULL;
122 }
```

Trace

```
strcpy(free_null_pointer_005_gbl_ptr,str)
```

02.wo_Defects/free_null_pointer.c:119

```
116 {  
117     char *str = "This is a string";  
118     free_null_pointer_005_func_001(strlen(str));  
  
119     strcpy(free_null_pointer_005_gbl_ptr,str);  
  
120     free(free_null_pointer_005_gbl_ptr);/* Tool should not detect this line as  
error */ /*No ERROR:Freeing a NULL pointer*/  
121     free_null_pointer_005_gbl_ptr = NULL;  
122 }
```

```
strcpy(free_null_pointer_005_gbl_ptr,str)
```

02.wo_Defects/free_null_pointer.c:119

```
116 {  
117     char *str = "This is a string";  
118     free_null_pointer_005_func_001(strlen(str));  
  
119     strcpy(free_null_pointer_005_gbl_ptr,str);  
  
120     free(free_null_pointer_005_gbl_ptr);/* Tool should not detect this line as  
error */ /*No ERROR:Freeing a NULL pointer*/  
121     free_null_pointer_005_gbl_ptr = NULL;  
122 }
```

02.wo_Defects/free_null_pointer.c:156

Level Medium

Status Not processed

```
153     {  
154         (s+i)->buf = (char*)malloc(25* sizeof(char));  
155     }  
  
156     strcpy((s+4)->buf,s1);
```

```
157 }  
158 if(free_null_pointer_006_func_001(flag)==0)  
159 {
```

Trace

```
strcpy((s+4)->buf,s1)
```

02.wo_Defects/free_null_pointer.c:156

```
153 {  
154     (s+i)->buf = (char*)malloc(25* sizeof(char));  
155 }
```

```
156 strcpy((s+4)->buf,s1);
```

```
157 }  
158 if(free_null_pointer_006_func_001(flag)==0)  
159 {
```

```
strcpy((s+4)->buf,s1)
```

02.wo_Defects/free_null_pointer.c:156

```
153 {  
154     (s+i)->buf = (char*)malloc(25* sizeof(char));  
155 }
```

```
156 strcpy((s+4)->buf,s1);
```

```
157 }  
158 if(free_null_pointer_006_func_001(flag)==0)  
159 {
```

02.wo_Defects/free_null_pointer.c:249

Level Medium

Status Not processed

```
246 char *str = "This is a string";
247 char *str1=NULL;
248 free_null_pointer_008_func_001(strlen(str),&str1);

249 strcpy(str1,str);

250 free(str1);/* Tool should not detect this line as error */ /*No ERROR:Freeing a NULL
pointer*/
251 str1 = NULL;
252 }
```

Trace

```
strcpy(str1,str)
```

02.wo_Defects/free_null_pointer.c:249

```
246 char *str = "This is a string";
247 char *str1=NULL;
248 free_null_pointer_008_func_001(strlen(str),&str1);

249 strcpy(str1,str);
```

```
250 free(str1);/* Tool should not detect this line as error */ /*No ERROR:Freeing
a NULL pointer*/
251 str1 = NULL;
252 }
```

```
strcpy(str1,str)
```

02.wo_Defects/free_null_pointer.c:249

```
246 char *str = "This is a string";
247 char *str1=NULL;
248 free_null_pointer_008_func_001(strlen(str),&str1);

249 strcpy(str1,str);
```

```
250 free(str1);/* Tool should not detect this line as error */ /*No ERROR:Freeing
a NULL pointer*/
251 str1 = NULL;
252 }
```

02.wo_Defects/free_null_pointer.c:284

Level Medium**Status** Not processed

```
281 free_null_pointer_009_func_001();
282 for(i=0;i<5;i++)
283 {
284     strcpy (free_null_pointer_009dst[i],"STRING");
285 }
286 while(1)
287 {
```

Trace

```
strcpy (free_null_pointer_009dst[i],"  
STRING")
```

02.wo_Defects/free_null_pointer.c:284

```
281 free_null_pointer_009_func_001();
282 for(i=0;i<5;i++)
283 {
284     strcpy (free_null_pointer_009dst[i],"STRING");
285 }
286 while(1)
287 {
```

```
strcpy (free_null_pointer_009dst[i],"  
STRING")
```

02.wo_Defects/free_null_pointer.c:284

```
281 free_null_pointer_009_func_001();
282 for(i=0;i<5;i++)
283 {
284     strcpy (free_null_pointer_009dst[i],"STRING");
```

```
285 }
286 while(1)
287 {
```

02.wo_Defects/free_null_pointer.c:325

Level Medium

Status Not processed

```
322 int i;
323 for(i=0;i<5;i++)
324 {
325     strcpy(*(dst+i),src[i]);
326 }
327 }
328 void free_null_pointer_010_func_003()
```

Trace

```
strcpy(*(dst+i),src[i])
```

02.wo_Defects/free_null_pointer.c:325

```
322 int i;
323 for(i=0;i<5;i++)
324 {
325     strcpy(*(dst+i),src[i]);
326 }
327 }
328 void free_null_pointer_010_func_003()
```

```
strcpy(*(dst+i),src[i])
```

02.wo_Defects/free_null_pointer.c:325

```
322 int i;
323 for(i=0;i<5;i++)
324 {
325     strcpy(*(dst+i),src[i]);
326 }
327 }
328 void free_null_pointer_010_func_003()
```

02.wo_Defects/invalid_memory_access.c:105

Level Medium

Status Not processed

```
102 buf = (char *) malloc (25 * sizeof(char));
103 if(buf!=NULL)
104 {
105     strcpy(buf,"This is String");
106     c = buf;
107     free(buf);
108     buf = NULL; /*Tool should not detect this line as error*/ /*No ERROR:
Invalid memory access to already freed area*/
```

Trace

```
strcpy(buf,"This is String")
```

02.wo_Defects/invalid_memory_access.c:105

```
102 buf = (char *) malloc (25 * sizeof(char));
103 if(buf!=NULL)
104 {
```

```
105 strcpy(buf,"This is String");

106     c = buf;
107     free(buf);
108         buf = NULL; /*Tool should not detect this line as error*/ /*No
ERROR:Invalid memory access to already freed area*/
```

```
strcpy(buf,"This is String")
```

02.wo_Defects/invalid_memory_access.c:105

```
102 buf = (char *) malloc (25 * sizeof(char));
103 if(buf!=NULL)
104 {

105 strcpy(buf,"This is String");

106     c = buf;
107     free(buf);
108         buf = NULL; /*Tool should not detect this line as error*/ /*No
ERROR:Invalid memory access to already freed area*/
```

02.wo_Defects/invalid_memory_access.c:213

Level Medium

Status Not processed

```
210
211 for(i=0;i<5;i++)
212 {
```

```
213 strcpy(*(ptr+i),"String"); /*Tool should not detect this line as error*/ /*No ERROR:
Invalid memory access to already freed area*/
```

```
214     free(ptr[i]);
215     ptr[i] = NULL;
216 }
```

Trace

```
strcpy(*(ptr+i),"String")
```

02.wo_Defects/invalid_memory_access.c:213

```
210
211 for(i=0;i<5;i++)
212 {
```

```
213 strcpy(*(ptr+i),"String"); /*Tool should not detect this line as error*/ /*No
ERROR:Invalid memory access to already freed area*/
```

```
214     free(ptr[i]);
215     ptr[i] = NULL;
216 }
```

```
strcpy(*(ptr+i),"String")
```

02.wo_Defects/invalid_memory_access.c:213

```
210
211 for(i=0;i<5;i++)
212 {
```

```
213 strcpy(*(ptr+i),"String"); /*Tool should not detect this line as error*/ /*No
ERROR:Invalid memory access to already freed area*/
```

```
214     free(ptr[i]);
215     ptr[i] = NULL;
216 }
```

02.wo_Defects/invalid_memory_access.c:567

Level Medium

Status Not processed

```
564         for(i=0;i<10;i++)
565     {
566             invalid_memory_access_016_doubleptr_gbl[i]=(char*) malloc
(10*sizeof(char));
```

```
567             strcpy(invalid_memory_access_016_doubleptr_gbl[i],"
```

```
STRING00");
```

```
568    }
569 }
570 }
```

Trace

```
strcpy
(invalid_memory_access_016_doubleptr_gb
l[i],"STRING00")
```

02.wo_Defects/invalid_memory_access.c:567

```
564      for(i=0;i<10;i++)
565  {
566      invalid_memory_access_016_doubleptr_gbl[i]=(char*) malloc
(10*sizeof(char));
567      strcpy(invalid_memory_access_016_doubleptr_gbl[i],""
STRING00");
568  }
569 }
570 }
```

```
strcpy
(invalid_memory_access_016_doubleptr_gb
l[i],"STRING00")
```

02.wo_Defects/invalid_memory_access.c:567

```
564      for(i=0;i<10;i++)
565  {
566      invalid_memory_access_016_doubleptr_gbl[i]=(char*) malloc
(10*sizeof(char));
567      strcpy(invalid_memory_access_016_doubleptr_gbl[i],""
STRING00");
568  }
569 }
570 }
```

02.wo_Defects/invalid_memory_access.c:576

Level Medium**Status** Not processed

```
573 {  
574     char s[10] ;  
575     printf("invalid gbl= %s \n",invalid_memory_access_016_doubleptr_gbl[0]);
```

576 strcpy(s,invalid_memory_access_016_doubleptr_gbl[0]);/*Tool should not detect
this line as error*/ /* No ERROR:Invalid memory access to already freed area*/

```
577     printf("invalid str= %s \n",s);  
578 }  
579
```

Trace

```
strcpy(s,  
invalid_memory_access_016_doubleptr_gbl  
[0])
```

02.wo_Defects/invalid_memory_access.c:576

```
573 {  
574     char s[10] ;  
575     printf("invalid gbl= %s \n",invalid_memory_access_016_doubleptr_gbl[0]);
```

576 strcpy(s,invalid_memory_access_016_doubleptr_gbl[0]);/*Tool should not
detect this line as error*/ /* No ERROR:Invalid memory access to already freed
area*/

```
577     printf("invalid str= %s \n",s);  
578 }  
579
```

```
strcpy(s,  
invalid_memory_access_016_doubleptr_gbl  
[0])
```

02.wo_Defects/invalid_memory_access.c:576

```
573 {  
574   char s[10] ;  
575   printf("invalid gbl= %s \n",invalid_memory_access_016_doubleptr_gbl[0]);  
  
576   strcpy(s,invalid_memory_access_016_doubleptr_gbl[0]);/*Tool should not  
detect this line as error*/ /* No ERROR:Invalid memory access to already freed  
area*/  
  
577   printf("invalid str= %s \n",s);  
578 }  
579
```

02.wo_Defects/invalid_memory_access.c:622

Level Medium

Status Not processed

```
619 {  
620   invalid_memory_access_017_doubleptr_gbl=(char*) malloc(10*sizeof(char));  
621   if(invalid_memory_access_017_doubleptr_gbl !=NULL)  
  
622   strcpy(invalid_memory_access_017_doubleptr_gbl,"TEST");  
  
623 }  
624  
625 void invalid_memory_access_017_func_003()
```

Trace

```
strcpy
(invalid_memory_access_017_doubleptr_gb
I,"TEST")
```

02.wo_Defects/invalid_memory_access.c:622

```
619 {
620 invalid_memory_access_017_doubleptr_gbl=(char*) malloc(10*sizeof(char));
621 if(invalid_memory_access_017_doubleptr_gbl !=NULL)

622 strcpy(invalid_memory_access_017_doubleptr_gbl,"TEST");

623 }
624
625 void invalid_memory_access_017_func_003()
```

```
strcpy
(invalid_memory_access_017_doubleptr_gb
I,"TEST")
```

02.wo_Defects/invalid_memory_access.c:622

```
619 {
620 invalid_memory_access_017_doubleptr_gbl=(char*) malloc(10*sizeof(char));
621 if(invalid_memory_access_017_doubleptr_gbl !=NULL)

622 strcpy(invalid_memory_access_017_doubleptr_gbl,"TEST");

623 }
624
625 void invalid_memory_access_017_func_003()
```

02.wo_Defects/invalid_memory_access.c:634

Level Medium

Status Not processed

```
631 {
632 char s[10] ;
633 printf("invalid gbl= %s \n",invalid_memory_access_017_doubleptr_gbl);
```

```
634 strcpy(s,invalid_memory_access_017_doubleptr_gbl); /*Tool should not detect this  
line as error*/ /*No ERROR:Invalid memory access to already freed area*/  
  
635 printf("invalid str= %s \n",s);  
636 }  
637
```

Trace

```
strcpy(s,  
invalid_memory_access_017_doubleptr_gbl  
)
```

02.wo_Defects/invalid_memory_access.c:634

```
631 {  
632 char s[10] ;  
633 printf("invalid gbl= %s \n",invalid_memory_access_017_doubleptr_gbl);
```

```
634 strcpy(s,invalid_memory_access_017_doubleptr_gbl); /*Tool should not  
detect this line as error*/ /*No ERROR:Invalid memory access to already freed  
area*/
```

```
635 printf("invalid str= %s \n",s);  
636 }  
637
```

```
strcpy(s,  
invalid_memory_access_017_doubleptr_gbl  
)
```

02.wo_Defects/invalid_memory_access.c:634

```
631 {  
632 char s[10] ;  
633 printf("invalid gbl= %s \n",invalid_memory_access_017_doubleptr_gbl);
```

```
634 strcpy(s,invalid_memory_access_017_doubleptr_gbl); /*Tool should not  
detect this line as error*/ /*No ERROR:Invalid memory access to already freed  
area*/
```

```
635 printf("invalid str= %s \n",s);  
636 }
```

637

02.wo_Defects/memory_allocation_failure.c:594

Level Medium**Status** Not processed

```
591 }  
592 else  
593 {  
  
594     strcpy( dptr[1],"STRING TEST" );  
  
595     if(1)  
596     {  
597         for (i = 0;i< 4; i++)
```

Trace

strcpy(dptr[1],"STRING TEST")

02.wo_Defects/memory_allocation_failure.c:594

```
591 }  
592 else  
593 {  
  
594     strcpy( dptr[1],"STRING TEST" );  
  
595     if(1)  
596     {  
597         for (i = 0;i< 4; i++)
```

```
strcpy( dptr[1],"STRING TEST" )
```

02.wo_Defects/memory_allocation_failure.c:594

```
591 }
592 else
593 {

594     strcpy( dptr[1],"STRING TEST" );

595     if(1)
596     {
597         for (i = 0;i< 4; i++)
```

02.wo_Defects/memory_leak.c:76

Level Medium

Status Not processed

```
73     char *str = "This is a string";
74     char *str1;
75     memory_leak_003_func_001(strlen(str),&str1); /*Tool should not detect this line as
error*/ /*No ERROR:Memory Leakage */

76     strcpy(str1,str);

77     free(str1);
78 }
79
```

Trace

```
strcpy(str1,str)
```

02.wo_Defects/memory_leak.c:76

```
73     char *str = "This is a string";
74     char *str1;
75     memory_leak_003_func_001(strlen(str),&str1); /*Tool should not detect this
line as error*/ /*No ERROR:Memory Leakage */
```

```
76 strcpy(str1,str);
```

```
77 free(str1);
78 }
79
```

```
strcpy(str1,str)
```

02.wo_Defects/memory_leak.c:76

```
73 char *str = "This is a string";
74 char *str1;
75 memory_leak_003_func_001(strlen(str),&str1); /*Tool should not detect this
line as error*/ /*No ERROR:Memory Leakage */
```

```
76 strcpy(str1,str);
```

```
77 free(str1);
78 }
79
```

02.wo_Defects/memory_leak.c:100

Level Medium

Status Not processed

```
97 {
98     (s+i)->buf = (char*)malloc(25* sizeof(char)); /*Tool should not detect this line as
error*/ /*No ERROR:Memory Leakage */
99 }
```

```
100 strcpy((s+4)->buf,s1);
```

```
101 for(i= 0; i<5 ;i++)
102     free((s+i)->buf);
103 free(s);
```

Trace

```
strcpy((s+4)->buf,s1)
```

02.wo_Defects/memory_leak.c:100

```
97 {  
98     (s+i)->buf = (char*)malloc(25* sizeof(char)); /*Tool should not detect this line  
as error*/ /*No ERROR:Memory Leakage */  
99 }  
  
100 strcpy((s+4)->buf,s1);  
  
101 for(i= 0; i<5 ;i++)  
102     free((s+i)->buf);  
103 free(s);
```

```
strcpy((s+4)->buf,s1)
```

02.wo_Defects/memory_leak.c:100

```
97 {  
98     (s+i)->buf = (char*)malloc(25* sizeof(char)); /*Tool should not detect this line  
as error*/ /*No ERROR:Memory Leakage */  
99 }  
  
100 strcpy((s+4)->buf,s1);  
  
101 for(i= 0; i<5 ;i++)  
102     free((s+i)->buf);  
103 free(s);
```

02.wo_Defects/memory_leak.c:277

Level Medium

Status Not processed

```
274     buf = (char *)calloc(50, sizeof(char)); /*Tool should not detect this line as error*/  
/*No ERROR:Memory Leakage */  
275 if(buf!=NULL)  
276 {  
  
277     strcpy(buf, "This Is A String");
```

```
278     un.u1 = buf;
279     char * buf = un.u1;
280     if (buf)
```

Trace

```
strcpy(buf, "This Is A String")
```

02.wo_Defects/memory_leak.c:277

```
274     buf = (char *)calloc(50, sizeof(char)); /*Tool should not detect this line as
error*/ /*No ERROR:Memory Leakage */
275 if(buf!=NULL)
276 {
```

```
277     strcpy(buf, "This Is A String");
```

```
278     un.u1 = buf;
279     char * buf = un.u1;
280     if (buf)
```

```
strcpy(buf, "This Is A String")
```

02.wo_Defects/memory_leak.c:277

```
274     buf = (char *)calloc(50, sizeof(char)); /*Tool should not detect this line as
error*/ /*No ERROR:Memory Leakage */
275 if(buf!=NULL)
276 {
```

```
277     strcpy(buf, "This Is A String");
```

```
278     un.u1 = buf;
279     char * buf = un.u1;
280     if (buf)
```

```
02.wo_Defects/memory_leak.c:408
```

Level Medium

Status Not processed

```
405 char *str1 = memory_leak_0015_func_001(strlen(str)); /*Tool should not detect this  
line as error*/ /*No ERROR:Memory Leakage */  
406 if(str1!=NULL)  
407 {  
  
408     strcpy(str1,str);  
  
409     free(str1);  
410 }  
411 }
```

Trace

strcpy(str1,str)

02.wo_Defects/memory_leak.c:408

```
405 char *str1 = memory_leak_0015_func_001(strlen(str)); /*Tool should not  
detect this line as error*/ /*No ERROR:Memory Leakage */  
406 if(str1!=NULL)  
407 {  
  
408     strcpy(str1,str);  
  
409     free(str1);  
410 }  
411 }
```

strcpy(str1,str)

02.wo_Defects/memory_leak.c:408

```
405 char *str1 = memory_leak_0015_func_001(strlen(str)); /*Tool should not  
detect this line as error*/ /*No ERROR:Memory Leakage */  
406 if(str1!=NULL)  
407 {  
  
408     strcpy(str1,str);
```

```
409     free(str1);
410 }
411 }
```

02.wo_Defects/memory_leak.c:431

Level Medium

Status Not processed

```
428 {
429     char *str = "This is a string";
430     memory_leak_0016_func_001(strlen(str));

431     strcpy(memory_leak_0016_gbl_ptr,str);

432     free(memory_leak_0016_gbl_ptr);
433     memory_leak_0016_gbl_ptr = NULL;
434 }
```

Trace

```
strcpy(memory_leak_0016_gbl_ptr,str)
```

02.wo_Defects/memory_leak.c:431

```
428 {
429     char *str = "This is a string";
430     memory_leak_0016_func_001(strlen(str));

431     strcpy(memory_leak_0016_gbl_ptr,str);

432     free(memory_leak_0016_gbl_ptr);
433     memory_leak_0016_gbl_ptr = NULL;
434 }
```

```
strcpy(memory_leak_0016_gbl_ptr,str)
```

02.wo_Defects/memory_leak.c:431

```
428 {  
429   char *str = "This is a string";  
430   memory_leak_0016_func_001(strlen(str));  
  
431   strcpy(memory_leak_0016_gbl_ptr,str);  
  
432   free(memory_leak_0016_gbl_ptr);  
433   memory_leak_0016_gbl_ptr = NULL;  
434 }
```

02.wo_Defects/memory_leak.c:526

Level Medium

Status Not processed

```
523   memory_leak_0018_func_001();  
524   for(i=0;i<5;i++)  
525   {  
  
526       strcpy (memory_leak_0018dst[i],"STRING");  
  
527   }  
528 while(1)  
529 {
```

Trace

```
strcpy (memory_leak_0018dst[i],"STRING")
```

02.wo_Defects/memory_leak.c:526

```
523   memory_leak_0018_func_001();  
524   for(i=0;i<5;i++)  
525   {  
  
526       strcpy (memory_leak_0018dst[i],"STRING");
```

```
527 }
528 while(1)
529 {
```

strcpy (memory_leak_0018dst[i],"STRING")

02.wo_Defects/memory_leak.c:526

```
523 memory_leak_0018_func_001();
524 for(i=0;i<5;i++)
525 {
526     strcpy (memory_leak_0018dst[i],"STRING");
527 }
528 while(1)
529 {
```

02.wo_Defects/null_pointer.c:259

Level Medium

Status Not processed

```
256 {
257 char *str = "This is a string";
258 null_pointer_015_func_001(strlen(str));

259 strcpy(null_pointer_015_gbl_ptr,str); /*Tool should not detect this line as error*/
/*NO ERROR:NULL pointer dereference*/

260 free(null_pointer_015_gbl_ptr);
261 null_pointer_015_gbl_ptr = NULL;
262 }
```

Trace

```
strcpy(null_pointer_015_gbl_ptr,str)
```

02.wo_Defects/null_pointer.c:259

```
256 {  
257   char *str = "This is a string";  
258   null_pointer_015_func_001(strlen(str));  
  
259   strcpy(null_pointer_015_gbl_ptr,str); /*Tool should not detect this line as  
error*/ /*NO ERROR:NULL pointer dereference*/  
  
260   free(null_pointer_015_gbl_ptr);  
261   null_pointer_015_gbl_ptr = NULL;  
262 }
```

```
strcpy(null_pointer_015_gbl_ptr,str)
```

02.wo_Defects/null_pointer.c:259

```
256 {  
257   char *str = "This is a string";  
258   null_pointer_015_func_001(strlen(str));  
  
259   strcpy(null_pointer_015_gbl_ptr,str); /*Tool should not detect this line as  
error*/ /*NO ERROR:NULL pointer dereference*/  
  
260   free(null_pointer_015_gbl_ptr);  
261   null_pointer_015_gbl_ptr = NULL;  
262 }
```

02.wo_Defects/null_pointer.c:354

Level Medium

Status Not processed

```
351   null_pointer_017_func_001();  
352   for(i=0;i<5;i++)  
353   {
```

```
354       strcpy (null_pointer_017dst[i],"STRING"); /*Tool should not detect this line  
as error*/ /*NO ERROR:NULL pointer dereference*/
```

```
355 }
356 while(1)
357 {
```

Trace

```
strcpy (null_pointer_017dst[i],"STRING")
```

02.wo_Defects/null_pointer.c:354

```
351 null_pointer_017_func_001();
352 for(i=0;i<5;i++)
353 {
```

```
354     strcpy (null_pointer_017dst[i],"STRING"); /*Tool should not detect
this line as error*/ /*NO ERROR:NULL pointer dereference*/
```

```
355 }
356 while(1)
357 {
```

```
strcpy (null_pointer_017dst[i],"STRING")
```

02.wo_Defects/null_pointer.c:354

```
351 null_pointer_017_func_001();
352 for(i=0;i<5;i++)
353 {
```

```
354     strcpy (null_pointer_017dst[i],"STRING"); /*Tool should not detect
this line as error*/ /*NO ERROR:NULL pointer dereference*/
```

```
355 }
356 while(1)
357 {
```

02.wo_Defects/st_underrun.c:81

Level Medium

Status Not processed

```
78 void st_underrun_003_func_001 (st_underrun_003_s_001 *s)
79 {
80     char buf[10] = "STRING";

81     strcpy(s->buf,buf);

82 }
83
84 void st_underrun_003_func_002 (st_underrun_003_s_001 *s)
```

Trace

strcpy(s->buf,buf)

02.wo_Defects/st_underrun.c:81

```
78 void st_underrun_003_func_001 (st_underrun_003_s_001 *s)
79 {
80     char buf[10] = "STRING";

81     strcpy(s->buf,buf);

82 }
83
84 void st_underrun_003_func_002 (st_underrun_003_s_001 *s)
```

strcpy(s->buf,buf)

02.wo_Defects/st_underrun.c:81

```
78 void st_underrun_003_func_001 (st_underrun_003_s_001 *s)
79 {
80     char buf[10] = "STRING";

81     strcpy(s->buf,buf);

82 }
83
84 void st_underrun_003_func_002 (st_underrun_003_s_001 *s)
```

02.wo_Defects/st_underrun.c:116

Level Medium**Status** Not processed

```
113 void st_underrun_004_func_002 (st_underrun_004_s_001 *s)
114 {
115     char buf[10] = "STRING";

116     strcpy(s->buf,buf);

117 }
118
119 st_underrun_004_s_001 st_underrun_004_func_001 (st_underrun_004_s_001 *s)
```

Trace

strcpy(s->buf,buf)

02.wo_Defects/st_underrun.c:116

```
113 void st_underrun_004_func_002 (st_underrun_004_s_001 *s)
114 {
115     char buf[10] = "STRING";

116     strcpy(s->buf,buf);

117 }
118
119 st_underrun_004_s_001 st_underrun_004_func_001
(st_underrun_004_s_001 *s)
```

```
strcpy(s->buf,buf)
```

02.wo_Defects/st_underrun.c:116

```
113 void st_underrun_004_func_002 (st_underrun_004_s_001 *s)
114 {
115     char buf[10] = "STRING";

116     strcpy(s->buf,buf);

117 }
118
119 st_underrun_004_s_001 st_underrun_004_func_001
(st_underrun_004_s_001 *s)
```

02.wo_Defects/st_underrun.c:213

Level Medium

Status Not processed

```
210     // JDR: this is an array overrun (copying 12 bytes into a 10-byte array)
211     // but maybe it is OK since the struct is guaranteed to have buf1 right
212     // after buf?
```

```
213 strcpy(s.buf,"STRING !!!!");
```

```
214 void (*func)(st_underrun_006_s_001);
215 func = st_underrun_006_func_001;
216 func(s);
```

Trace

```
strcpy(s.buf,"STRING !!!!")
```

02.wo_Defects/st_underrun.c:213

```
210     // JDR: this is an array overrun (copying 12 bytes into a 10-byte array)
211     // but maybe it is OK since the struct is guaranteed to have buf1 right
212     // after buf?
```

```
213 strcpy(s.buf,"STRING !!!!");  
  
214 void (*func)(st_underrun_006_s_001);  
215 func = st_underrun_006_func_001;  
216 func(s);
```

strcpy(s.buf,"STRING !!!")

02.wo_Defects/st_underrun.c:213

```
210     // JDR: this is an array overrun (copying 12 bytes into a 10-byte array)  
211     // but maybe it is OK since the struct is guaranteed to have buf1 right  
212     // after buf?
```

```
213 strcpy(s.buf,"STRING !!!!");
```

```
214 void (*func)(st_underrun_006_s_001);  
215 func = st_underrun_006_func_001;  
216 func(s);
```

02.wo_Defects/uninit_memory_access.c:55

Level Medium

Status Not processed

```
52 char *str2 = "THIS IS STRING";  
53 if (str1!=NULL)  
54 {  
  
55     strcpy(str1, str2);  
  
56     printf("%s %s\n",str1,str2); /*Tool should not detect this line as error*/ /*No  
ERROR:Uninitialized Memory Access*/  
57     free(str1);  
58 }
```

Trace

strcpy(str1, str2)

02.wo_Defects/uninit_memory_access.c:55

```
52 char *str2 = "THIS IS STRING";
53 if (str1!=NULL)
54 {
55     strcpy(str1, str2);

56     printf("%s %s\n",str1,str2); /*Tool should not detect this line as error*/ /*No
ERROR:Uninitialized Memory Access*/
57     free(str1);
58 }
```

strcpy(str1, str2)

02.wo_Defects/uninit_memory_access.c:55

```
52 char *str2 = "THIS IS STRING";
53 if (str1!=NULL)
54 {
55     strcpy(str1, str2);

56     printf("%s %s\n",str1,str2); /*Tool should not detect this line as error*/ /*No
ERROR:Uninitialized Memory Access*/
57     free(str1);
58 }
```

02.wo_Defects/uninit_pointer.c:197

Level Medium

Status Not processed

```
194 }
195 if(uninit_pointer_009_func_001(flag)>0)
196 {
```

197 strcpy(buf1,buf); /*Tool should not detect this line as error*/ /*No ERROR:
Uninitialized pointer*/

```
198  }
199 }
200
```

Trace

```
strcpy(buf1,buf)
```

02.wo_Defects/uninit_pointer.c:197

```
194  }
195  if(uninit_pointer_009_func_001(flag)>0)
196  {

197      strcpy(buf1,buf); /*Tool should not detect this line as error*/ /*No
ERROR:Uninitialized pointer*/

198  }
199 }
200
```

```
strcpy(buf1,buf)
```

02.wo_Defects/uninit_pointer.c:197

```
194  }
195  if(uninit_pointer_009_func_001(flag)>0)
196  {

197      strcpy(buf1,buf); /*Tool should not detect this line as error*/ /*No
ERROR:Uninitialized pointer*/

198  }
199 }
200
```

02.wo_Defects/uninit_pointer.c:400

Level Medium

Status Not processed

```
397     for(i=0;i<10;i++)
398     {
399         uninit_pointer_016_gbl_doubleptr[i]=(char*) malloc(10*sizeof(char));
400
401     }
402 }
403 }
```

Trace

```
strcpy(uninit_pointer_016_gbl_doubleptr[i],"  
STRING00")
```

02.wo_Defects/uninit_pointer.c:400

```
397     for(i=0;i<10;i++)
398     {
399         uninit_pointer_016_gbl_doubleptr[i]=(char*) malloc(10*sizeof
(char));
400
401     }
402 }
403 }
```

```
strcpy(uninit_pointer_016_gbl_doubleptr[i],"  
STRING00")
```

02.wo_Defects/uninit_pointer.c:400

```
397     for(i=0;i<10;i++)
398     {
399         uninit_pointer_016_gbl_doubleptr[i]=(char*) malloc(10*sizeof
(char));
400
401     }
402 }
403 }
```

```
401    }
402 }
403 }
```

02.wo_Defects/uninit_pointer.c:423

Level Medium

Status Not processed

```
420 if(i==7)
421 {
422   printf("unint p %s \n",uninit_pointer_016_gbl_doubleptr[i]);
423   strcpy(s,uninit_pointer_016_gbl_doubleptr[i]); /*Tool should not detect this line as
error*/ /*No ERROR:Uninitialized pointer*/
424   printf("unint p %s \n",s);
425 }
426   free (uninit_pointer_016_gbl_doubleptr[i]);
```

Trace

```
strcpy(s,uninit_pointer_016_gbl_doubleptr[i])
```

02.wo_Defects/uninit_pointer.c:423

```
420 if(i==7)
421 {
422   printf("unint p %s \n",uninit_pointer_016_gbl_doubleptr[i]);
423   strcpy(s,uninit_pointer_016_gbl_doubleptr[i]); /*Tool should not detect this
line as error*/ /*No ERROR:Uninitialized pointer*/
424   printf("unint p %s \n",s);
425 }
426   free (uninit_pointer_016_gbl_doubleptr[i]);
```

```
strcpy(s,uninit_pointer_016_gbl_doubleptr[i])
```

02.wo_Defects/uninit_pointer.c:423

```
420 if(i==7)
421 {
422   printf("unint p %s \n",uninit_pointer_016_gbl_doubleptr[i]);

423 strcpy(s,uninit_pointer_016_gbl_doubleptr[i]); /*Tool should not detect this
line as error*/ /*No ERROR:Uninitialized pointer*/

424   printf("unint p %s \n",s);
425 }
426 free (uninit_pointer_016_gbl_doubleptr[i]);
```

02.wo_Defects/uninit_var.c:152

Level Medium

Status Not processed

```
149 void uninit_var_009_func_001 (char buf[])
150 {
151   char ret[] = "This is a string";

152 strcpy(buf,ret); /*Tool should not detect this line as error*/ /*No ERROR:Uninitialized
Variable*/

153 }
154
155 void uninit_var_009 ()
```

Trace

```
strcpy(buf,ret)
```

02.wo_Defects/uninit_var.c:152

```
149 void uninit_var_009_func_001 (char buf[])
150 {
151   char ret[] = "This is a string";
```

```
152 strcpy(buf,ret); /*Tool should not detect this line as error*/ /*No ERROR:  
Uninitialized Variable*/
```

```
153 }  
154  
155 void uninit_var_009 ()
```

```
strcpy(buf,ret)
```

02.wo_Defects/uninit_var.c:152

```
149 void uninit_var_009_func_001 (char buf[])  
150 {  
151   char ret[] = "This is a string";
```

```
152 strcpy(buf,ret); /*Tool should not detect this line as error*/ /*No ERROR:  
Uninitialized Variable*/
```

```
153 }  
154  
155 void uninit_var_009 ()
```

02.wo_Defects/wrong_arguments_func_pointer.c:208

Level Medium

Status Not processed

```
205 */  
206 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2, char*str3)  
207 {  
  
208   strcpy(str1,str2);  
  
209   strcpy(str3,str1);  
210   return ('c');  
211 }
```

Trace

```
strcpy(str1,str2)
```

02.wo_Defects/wrong_arguments_func_pointer.c:208

```
205 */  
206 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2,  
char*str3)  
207 {  
  
208     strcpy(str1,str2);  
  
209     strcpy(str3,str1);  
210     return ('c');  
211 }
```

```
strcpy(str1,str2)
```

02.wo_Defects/wrong_arguments_func_pointer.c:208

```
205 */  
206 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2,  
char*str3)  
207 {  
  
208     strcpy(str1,str2);  
  
209     strcpy(str3,str1);  
210     return ('c');  
211 }
```

02.wo_Defects/wrong_arguments_func_pointer.c:209

Level Medium

Status Not processed

```
206 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2, char*str3)  
207 {  
208     strcpy(str1,str2);  
  
209     strcpy(str3,str1);
```

```
210     return ('c');
211 }
212
```

Trace

```
strcpy(str3,str1)
```

02.wo_Defects/wrong_arguments_func_pointer.c:209

```
206 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2,
char*str3)
207 {
208     strcpy(str1,str2);
```

```
209     strcpy(str3,str1);
```

```
210     return ('c');
211 }
212
```

```
strcpy(str3,str1)
```

02.wo_Defects/wrong_arguments_func_pointer.c:209

```
206 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2,
char*str3)
207 {
208     strcpy(str1,str2);
```

```
209     strcpy(str3,str1);
```

```
210     return ('c');
211 }
212
```

02.wo_Defects/wrong_arguments_func_pointer.c:381

Level Medium

Status Not processed

```
378 void (*fptr)(int,char **);
379 fptr = wrong_arguments_func_pointer_013_func_001;
380 fptr(strlen(str),&str1); /*Tool should not detect this line as error**/*No ERROR:
Wrong arguments passed to a function pointer*/

381 strcpy(str1,str);

382 free(str1);
383 str1 = NULL;
384 }
```

Trace

```
strcpy(str1,str)
```

02.wo_Defects/wrong_arguments_func_pointer.c:381

```
378 void (*fptr)(int,char **);
379 fptr = wrong_arguments_func_pointer_013_func_001;
380 fptr(strlen(str),&str1); /*Tool should not detect this line as error**/*No
ERROR:Wrong arguments passed to a function pointer*/

381 strcpy(str1,str);

382 free(str1);
383 str1 = NULL;
384 }
```

```
strcpy(str1,str)
```

02.wo_Defects/wrong_arguments_func_pointer.c:381

```
378 void (*fptr)(int,char **);
379 fptr = wrong_arguments_func_pointer_013_func_001;
380 fptr(strlen(str),&str1); /*Tool should not detect this line as error**/*No
ERROR:Wrong arguments passed to a function pointer*/

381 strcpy(str1,str);

382 free(str1);
383 str1 = NULL;
384 }
```

02.wo_Defects/wrong_arguments_func_pointer.c:435

Level Medium**Status** Not processed

```
432 int i;
433 for(i=0;i<5;i++)
434 {
435     strcpy(*(wrong_arguments_func_pointer_015_dst1_gbl+i),src[i]);
436 }
437 }
438
```

Trace

```
strcpy(*  
(wrong_arguments_func_pointer_015_dst1_  
gbl+i),src[i])
```

02.wo_Defects/wrong_arguments_func_pointer.c:435

```
432 int i;
433 for(i=0;i<5;i++)
434 {
435     strcpy(*(wrong_arguments_func_pointer_015_dst1_gbl+i),src[i]);
436 }
437 }
438
```

```
strcpy(*  
(wrong_arguments_func_pointer_015_dst1_  
gbl+i),src[i])
```

02.wo_Defects/wrong_arguments_func_pointer.c:435

```
432 int i;
433 for(i=0;i<5;i++)
434 {
```

```
435     strcpy(*(wrong_arguments_func_pointer_015_dst1_gbl+i),src[i]);  
  
436 }  
437 }  
438
```

02.wo_Defects/wrong_arguments_func_pointer.c:474

Level Medium

Status Not processed

```
471 char wrong_arguments_func_pointer_016_func_001(char *str1, int *str2, float*str3)  
472 {  
473     char s[20];  
  
474     strcpy(s,str1);  
  
475     *str2 +=1;  
476     *str3 +=1;  
477     return (*str2);
```

Trace

strcpy(s,str1)

02.wo_Defects/wrong_arguments_func_pointer.c:474

```
471 char wrong_arguments_func_pointer_016_func_001(char *str1, int *str2,  
float*str3)  
472 {  
473     char s[20];  
  
474     strcpy(s,str1);  
  
475     *str2 +=1;  
476     *str3 +=1;  
477     return (*str2);
```

```
strcpy(s,str1)
```

02.wo_Defects/wrong_arguments_func_pointer.c:474

```
471 char wrong_arguments_func_pointer_016_func_001(char *str1, int *str2,
float*str3)
472 {
473     char s[20];

474     strcpy(s,str1);

475     *str2 +=1;
476     *str3 +=1;
477     return (*str2);
```

Invalid memory deallocation function usage (C/C++)

Description

Invalid arguments for the memory deallocation function.

Example

The following example shows an invalid memory deallocation function usage.

```
int *buf = (int *)malloc(4 * sizeof(int));
++buf;
free(buf);
```

A safe alternative.

```
int *buf = (int *)malloc(4 * sizeof(int));
int *buf2 = buf + 1;
free(buf);
```

Recommendations

- Correctly deallocated allocated memory.

Links

1. free — cppreference.com

Vulnerability Entries

01.w_Defects/free_nondynamically_allocated_memory.c:22

Level Medium

Status Not processed

```
19 {  
20   char* ptr="a";  
21   if(1)  
  
22   free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/  
  
23  
24 }  
25
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;  
277 void free_nondynamic_allocated_memory_main ()  
278 {  
  
279   if (vflag == 1 || vflag ==888)  
  
280   {  
281     free_nondynamic_allocated_memory_001();  
282   }
```

```
free(ptr); /*Tool should detect this line as  
error*/ /*ERROR:Free memory not allocated  
dynamically*/
```

01.w_Defects/free_nondynamically_allocated_memory.c:22

```
19 {  
20   char* ptr="a";  
21   if(1)
```

```
22   free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/
```

```
23  
24 }  
25
```

01.w_Defects/free_nondynamically_allocated_memory.c:36

Level Medium

Status Not processed

```
33   char* ptr="a";  
34   int flag=1;  
35   if(flag>0)
```

```
36   free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/
```

```
37 }  
38  
39 /*
```

Trace

```
vflag == 1
```

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;
277 void free_nondynamic_allocated_memory_main ()
278 {
279     if (vflag == 1 || vflag == 888)
280     {
281         free_nondynamic_allocated_memory_001();
282     }
}
```

free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/

01.w_Defects/free_nondynamically_allocated_memory.c:36

```
33     char* ptr = "a";
34     int flag = 1;
35     if(flag > 0)
36         free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not
37     allocated dynamically*/
38
39 /*
```

01.w_Defects/free_nondynamically_allocated_memory.c:48

Level Medium

Status Not processed

```
45 void free_nondynamic_allocated_memory_003()
46 {
47     free_nondynamic_allocated_memory_003_gbl_ptr = "STRING"; /*Tool should
detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/
```

```
48     free(free_nondynamic_allocated_memory_003_gbl_ptr);
```

```
49 }
```

```
50
```

```
51
```

Trace

```
vflag == 1
```

```
01.w_Defects/free_nondynamically_allocated_memory.c:279
```

```
276 extern volatile int vflag;
```

```
277 void free_nondynamic_allocated_memory_main ()
```

```
278 {
```

```
279     if (vflag == 1 || vflag ==888)
```

```
280     {
```

```
281         free_nondynamic_allocated_memory_001();
```

```
282     }
```

```
free
```

```
(free_nondynamic_allocated_memory_003_
```

```
gbl_ptr)
```

```
01.w_Defects/free_nondynamically_allocated_memory.c:48
```

```
45 void free_nondynamic_allocated_memory_003()
```

```
46 {
```

```
47     free_nondynamic_allocated_memory_003_gbl_ptr = "STRING"; /*Tool  
should detect this line as error*/ /*ERROR:Free memory not allocated  
dynamically*/
```

```
48     free(free_nondynamic_allocated_memory_003_gbl_ptr);
```

```
49 }
```

```
50
```

```
51
```

01.w_Defects/free_nondynamically_allocated_memory.c:62

Level Medium**Status** Not processed

```
59 char** ptr;
60 char *ptr1 = "STRING";
61 ptr = &ptr1;

62 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not
allocated dynamically*/

63 }
64
65 /*
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;
277 void free_nondynamic_allocated_memory_main ()
278 {

279 if (vflag == 1 || vflag ==888)

280 {
281     free_nondynamic_allocated_memory_001();
282 }
```

```
free(ptr); /*Tool should detect this line as
error*/ /*ERROR:Free memory not allocated
dynamically*/
```

01.w_Defects/free_nondynamically_allocated_memory.c:62

```
59 char** ptr;
60 char *ptr1 = "STRING";
61 ptr = &ptr1;
```

```
62 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/  
  
63 }  
64  
65 /*
```

01.w_Defects/free_nondynamically_allocated_memory.c:86

Level Medium

Status Not processed

```
83 }  
84 free(buf1);  
85 free(buf2);
```

```
86 free(pbuf); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/
```

```
87 }  
88  
89 /*
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;  
277 void free_nondynamic_allocated_memory_main ()  
278 {
```

```
279 if (vflag == 1 || vflag ==888)
```

```
280 {  
281     free_nondynamic_allocated_memory_001();  
282 }
```

```
free(pbuf); /*Tool should detect this line as  
error*/ /*ERROR:Free memory not allocated  
dynamically*/
```

01.w_Defects/free_nondynamically_allocated_memory.c:86

```
83 }  
84 free(buf1);  
85 free(buf2);
```

```
86 free(pbuf); /*Tool should detect this line as error*/ /*ERROR:Free memory  
not allocated dynamically*/
```

```
87 }  
88  
89 /*
```

01.w_Defects/free_nondynamically_allocated_memory.c:103

Level Medium

Status Not processed

```
100 double* ptr4=&d;  
101  
102
```

```
103 free(ptr4); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/
```

```
104 }  
105  
106 /*
```

Trace

```
vflag == 1
```

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;
277 void free_nondynamic_allocated_memory_main ()
278 {
279   if (vflag == 1 || vflag ==888)
280   {
281     free_nondynamic_allocated_memory_001();
282 }
```

free(ptr4); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/

01.w_Defects/free_nondynamically_allocated_memory.c:103

```
100 double* ptr4=&d;
101
102
103 free(ptr4); /*Tool should detect this line as error*/ /*ERROR:Free memory
not allocated dynamically*/
104 }
105
106 /*
```

01.w_Defects/free_nondynamically_allocated_memory.c:115

Level Medium

Status Not processed

```
112 {
113   char* ptr="a";
114   while(1)
```

115 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not

allocated dynamically*/

```
116 }  
117  
118 /*
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;  
277 void free_nondynamic_allocated_memory_main ()  
278 {  
  
279 if (vflag == 1 || vflag ==888)  
  
280 {  
281     free_nondynamic_allocated_memory_001();  
282 }
```

free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/

01.w_Defects/free_nondynamically_allocated_memory.c:115

```
112 {  
113     char* ptr="a";  
114     while(1)  
  
115     free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/  
  
116 }  
117  
118 /*
```

01.w_Defects/free_nondynamically_allocated_memory.c:128

Level Medium**Status** Not processed

```
125 char* ptr="a";
126 int a=2,b=0;
127 while(a>b)
```

128 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/

```
129 }
130
131 /*
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;
277 void free_nondynamic_allocated_memory_main ()
278 {
```

279 if (vflag == 1 || vflag ==888)

```
280 {
281     free_nondynamic_allocated_memory_001();
282 }
```

free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/

01.w_Defects/free_nondynamically_allocated_memory.c:128

```
125 char* ptr="a";
126 int a=2,b=0;
127 while(a>b)
```

```
128 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/
```

```
129 }  
130  
131 /*
```

01.w_Defects/free_nondynamically_allocated_memory.c:141

Level Medium

Status Not processed

```
138 char* ptr="s";  
139 int a=0,b=2;  
140 while(a<b)
```

```
141 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/
```

```
142 }  
143  
144 /*
```

Trace

```
vflag == 1
```

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;  
277 void free_nondynamic_allocated_memory_main ()  
278 {
```

```
279 if (vflag == 1 || vflag ==888)
```

```
280 {  
281     free_nondynamic_allocated_memory_001();  
282 }
```

```
free(ptr); /*Tool should detect this line as  
error*/ /*ERROR:Free memory not allocated  
dynamically*/
```

01.w_Defects/free_nondynamically_allocated_memory.c:141

```
138 char* ptr="s";  
139 int a=0,b=2;  
140 while(a<b)
```

```
141 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/
```

```
142 }  
143  
144 /*
```

01.w_Defects/free_nondynamically_allocated_memory.c:155

Level Medium

Status Not processed

```
152 for(i=0;i<5;i++)  
153 {  
154     char* ptr="k";
```

```
155         free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/
```

```
156 }  
157 }  
158
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;
277 void free_nondynamic_allocated_memory_main ()
278 {
279     if (vflag == 1 || vflag == 888)
280     {
281         free_nondynamic_allocated_memory_001();
282     }
```

free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/

01.w_Defects/free_nondynamically_allocated_memory.c:155

```
152     for(i=0;i<5;i++)
153     {
154         char* ptr="k";
155         free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/
156     }
157 }
158
```

01.w_Defects/free_nondynamically_allocated_memory.c:170

Level Medium

Status Not processed

```
167     for(i=0;i<5;i++)
168     {
169         char* ptr="l";
170         free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not
```

allocated dynamically*/

```
171 }
172 }
173
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;
277 void free_nondynamic_allocated_memory_main ()
278 {
279     if (vflag == 1 || vflag ==888)
280     {
281         free_nondynamic_allocated_memory_001();
282     }
```

free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/

01.w_Defects/free_nondynamically_allocated_memory.c:170

```
167     for(i=0;i<5;i++)
168     {
169         char* ptr="I";
170         free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/
171     }
172 }
173
```

01.w_Defects/free_nondynamically_allocated_memory.c:187

Level Medium**Status** Not processed

```
184 {  
185     a++;  
186     if(a==1)  
  
187         free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free memory not  
allocated dynamically*/  
  
188 }  
189 }  
190
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;  
277 void free_nondynamic_allocated_memory_main ()  
278 {  
  
279     if (vflag == 1 || vflag ==888)  
  
280     {  
281         free_nondynamic_allocated_memory_001();  
282     }
```

```
free(ptr); /*Tool should detect this line as  
error*/ /*ERROR:Free memory not allocated  
dynamically*/
```

01.w_Defects/free_nondynamically_allocated_memory.c:187

```
184 {  
185     a++;  
186     if(a==1)
```

```
187     free(ptr); /*Tool should detect this line as error*/ /*ERROR:Free  
memory not allocated dynamically*/  
  
188 }  
189 }  
190
```

01.w_Defects/free_nondynamically_allocated_memory.c:209

Level Medium

Status Not processed

```
206 free_nondynamic_allocated_memory_struct_013* new_struct=malloc(sizeof  
(free_nondynamic_allocated_memory_struct_013));  
207 free_nondynamic_allocated_memory_struct_013 str;  
208 new_struct->next = &str;
```

```
209 free(new_struct->next);/*Tool should detect this line as error*/ /*ERROR:Free  
memory not allocated dynamically*/
```

```
210 free(new_struct);  
211 }  
212
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;  
277 void free_nondynamic_allocated_memory_main ()  
278 {
```

```
279 if (vflag == 1 || vflag ==888)
```

```
280 {  
281     free_nondynamic_allocated_memory_001();  
282 }
```

new_struct->next

01.w_Defects/free_nondynamically_allocated_memory.c:209

```
206 free_nondynamic_allocated_memory_struct_013* new_struct=malloc(sizeof(free_nondynamic_allocated_memory_struct_013));  
207 free_nondynamic_allocated_memory_struct_013 str;  
208 new_struct->next = &str;  
  
209 free(new_struct->next);/*Tool should detect this line as error*/ /*ERROR:  
Free memory not allocated dynamically*/  
  
210 free(new_struct);  
211 }  
212
```

01.w_Defects/free_nondynamically_allocated_memory.c:229

Level Medium**Status** Not processed

```
226 {  
227 free_nondynamic_allocated_memory_struct_014  
free_nondynamic_allocated_memory_st ;  
228 free_nondynamic_allocated_memory_str =  
&free_nondynamic_allocated_memory_st;  
  
229 free(free_nondynamic_allocated_memory_str);/*Tool should detect this line as  
error*/ /*ERROR:Free memory not allocated dynamically*/  
  
230 }  
231  
232 /*
```

Trace

```
vflag == 1
```

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;
277 void free_nondynamic_allocated_memory_main ()
278 {
279     if (vflag == 1 || vflag ==888)
280     {
281         free_nondynamic_allocated_memory_001();
282     }
```

```
free
(free_nondynamic_allocated_memory_str);
/*Tool should detect this line as error*/
/*ERROR:Free memory not allocated
dynamically*/
```

01.w_Defects/free_nondynamically_allocated_memory.c:229

```
226 {
227     free_nondynamic_allocated_memory_struct_014
free_nondynamic_allocated_memory_st ;
228     free_nondynamic_allocated_memory_str =
&free_nondynamic_allocated_memory_st;

229     free(free_nondynamic_allocated_memory_str);/*Tool should detect this line
as error*/ /*ERROR:Free memory not allocated dynamically*/

230 }
231
232 /*
```

01.w_Defects/free_nondynamically_allocated_memory.c:239

Level Medium

Status Not processed

```
236 char *free_nondynamic_allocated_memory_015_gbl_ptr;
237 void free_nondynamic_allocated_memory_015_func_001()
```

```
238 {  
  
239 free(free_nondynamic_allocated_memory_015_gbl_ptr); /*Tool should detect this  
line as error*/ /*ERROR:Free memory not allocated dynamically*/  
  
240 }  
241  
242 void free_nondynamic_allocated_memory_015()
```

Trace

vflag == 1

```
01.w_Defects/free_nondynamically_allocated_memory.c:279  
  
276 extern volatile int vflag;  
277 void free_nondynamic_allocated_memory_main ()  
278 {  
  
279 if (vflag == 1 || vflag ==888)  
  
280 {  
281     free_nondynamic_allocated_memory_001();  
282 }
```

free
(free_nondynamic_allocated_memory_015_
gbl_ptr); /*Tool should detect this line as
error*/ /*ERROR:Free memory not allocated
dynamically*/

```
01.w_Defects/free_nondynamically_allocated_memory.c:239
```

```
236 char *free_nondynamic_allocated_memory_015_gbl_ptr;  
237 void free_nondynamic_allocated_memory_015_func_001()  
238 {  
  
239 free(free_nondynamic_allocated_memory_015_gbl_ptr); /*Tool should  
detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/  
  
240 }  
241  
242 void free_nondynamic_allocated_memory_015()
```

01.w_Defects/free_nondynamically_allocated_memory.c:262

Level Medium**Status** Not processed

```
259 void free_nondynamic_allocated_memory_016_func_002()
260 {
261     if(free_nondynamic_allocated_memory_016_gbl_var =='A')

262     free(free_nondynamic_allocated_memory_016_gbl_ptr);/*Tool should detect this
line as error*/ /*ERROR:Free memory not allocated dynamically*/

263 }
264
265 void free_nondynamic_allocated_memory_016()
```

Trace

vflag == 1

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;
277 void free_nondynamic_allocated_memory_main ()
278 {

279     if (vflag == 1 || vflag ==888)

280     {
281         free_nondynamic_allocated_memory_001();
282     }
```

```
free  
(free_nondynamic_allocated_memory_016_  
gbl_ptr);/*Tool should detect this line as  
error*/ /*ERROR:Free memory not allocated  
dynamically*/
```

01.w_Defects/free_nondynamically_allocated_memory.c:262

```
259 void free_nondynamic_allocated_memory_016_func_002()  
260 {  
261   if(free_nondynamic_allocated_memory_016_gbl_var =='A')  
  
262     free(free_nondynamic_allocated_memory_016_gbl_ptr);/*Tool should  
detect this line as error*/ /*ERROR:Free memory not allocated dynamically*/  
  
263 }  
264  
265 void free_nondynamic_allocated_memory_016()
```

01.w_Defects/uninit_memory_access.c:420

Level Medium

Status Not processed

```
417 if(p != NULL)  
418 {  
419   ret = p->b; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Memory  
Access*/  
  
420   free(p);  
  
421   p= NULL;  
422 }  
423 }
```

Trace

```
vflag == 1
```

```
01.w_Defects/uninit_memory_access.c:462
```

```
459 extern volatile int vflag;
460 void uninit_memory_access_main ()
461 {
462     if (vflag == 1 || vflag == 888)
463     {
464         uninit_memory_access_001();
465     }
}
```

```
free(p)
```

```
01.w_Defects/uninit_memory_access.c:420
```

```
417     if (p != NULL)
418     {
419         ret = p->b; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
Memory Access*/
420     free(p);
421     p = NULL;
422     }
423 }
```

Loss of precision (C/C++)

Description

Loss of precision can occur when:

1. A negative value is implicitly converted to an unsigned value in an assignment, comparison or multiplication.
2. Assignment/initialization when source value is greater than the max value of target.

Example

In the following example a negative value is implicitly converted to an unsigned value:

```
void addAssign() {  
    unsigned long L = 1000;  
    int I = -100;  
    U8 += L;  
}  
  
void mulAssign() {  
    unsigned long L = 1000;  
    int I = -1;  
    L *= I;  
}
```

Recommendations

- If possible, avoid conversions with numeric types.
- Always check the valid ranges.

Links

1. Loss of significance — Wikipedia
2. Precision of floating point numbers in C++
3. CWE-682: Incorrect Calculation

Vulnerability Entries

01.w_Defects/bit_shift.c:35

Level Medium

Status Not processed

```
32 long a = 1;  
33 long ret;  
34 ret = a << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/  
  
35     sink = ret;  
  
36 }  
37  
38 /*
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;
246 void bit_shift_main ()
247 {
```

```
248 if (vflag == 1 || vflag ==888)
```

```
249 {
250     bit_shift_001();
251 }
```

sink = ret

01.w_Defects/bit_shift.c:35

```
32 long a = 1;
33 long ret;
34 ret = a << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift
error*/
```

```
35     sink = ret;
```

```
36 }
37
38 /*
```

01.w_Defects/bit_shift.c:59

Level Medium

Status Not processed

```
56 unsigned long a = 1;
57 unsigned long ret;
58 ret = a << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/
```

```
59     sink = ret;
```

```
60 }  
61  
62 /*
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;  
246 void bit_shift_main ()  
247 {
```

```
248 if (vflag == 1 || vflag ==888)
```

```
249 {  
250     bit_shift_001();  
251 }
```

sink = ret

01.w_Defects/bit_shift.c:59

```
56     unsigned long a = 1;  
57     unsigned long ret;  
58     ret = a << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift  
error*/
```

```
59     sink = ret;
```

```
60 }  
61  
62 /*
```

01.w_Defects/data_lost.c:22

Level Medium

Status Not processed

```
19 {  
20   char ret;  
21   short a = 0x80;  
  
22   ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost  
because of cast*/  
  
23   sink = ret;  
24 }  
25
```

Trace

```
vflag ==1
```

```
01.w_Defects/data_lost.c:267
```

```
264 extern volatile int vflag;  
265 void data_lost_main ()  
266 {
```

```
267   if (vflag ==1 || vflag ==888)
```

```
268   {  
269     data_lost_001();  
270   }
```

```
ret = a; /*Tool should detect this line as  
error*/ /*ERROR:Integer precision lost  
because of cast*/
```

```
01.w_Defects/data_lost.c:22
```

```
19 {  
20   char ret;  
21   short a = 0x80;
```

```
22   ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision  
lost because of cast*/
```

```
23   sink = ret;
```

```
24 }  
25
```

01.w_Defects/data_lost.c:34

Level Medium

Status Not processed

```
31 {  
32     short ret;  
33     int a = 0x8000;  
  
34     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost  
because of cast*/  
  
35     sink = ret;  
36 }  
37
```

Trace

vflag ==1

01.w_Defects/data_lost.c:267

```
264 extern volatile int vflag;  
265 void data_lost_main ()  
266 {  
  
267     if (vflag ==1 || vflag ==888)  
  
268     {  
269         data_lost_001();  
270     }
```

ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

01.w_Defects/data_lost.c:34

```
31 {  
32     short ret;  
33     int a = 0x8000;  
  
34     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/  
  
35     sink = ret;  
36 }  
37
```

01.w_Defects/data_lost.c:46

Level Medium

Status Not processed

```
43 {  
44     short ret;  
45     long a = 0x8000;  
  
46     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/  
  
47     sink = ret;  
48 }  
49
```

Trace

vflag ==1

01.w_Defects/data_lost.c:267

```
264 extern volatile int vflag;
265 void data_lost_main ()
266 {
```

```
267 if (vflag ==1 || vflag ==888)
```

```
268 {
269     data_lost_001();
270 }
```

ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

01.w_Defects/data_lost.c:46

```
43 {
44     short ret;
45     long a = 0x8000;
```

```
46     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/
```

```
47     sink = ret;
48 }
49
```

01.w_Defects/data_lost.c:94

Level Medium

Status Not processed

```
91 {
92     unsigned char ret;
93     unsigned short a = 0x0100;
```

```
94     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost
```

because of cast*/

```
95     sink = ret;  
96 }  
97
```

Trace

vflag ==1

01.w_Defects/data_lost.c:267

```
264 extern volatile int vflag;  
265 void data_lost_main ()  
266 {  
  
267 if (vflag ==1 || vflag ==888)  
  
268 {  
269     data_lost_001();  
270 }
```

ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

01.w_Defects/data_lost.c:94

```
91 {  
92     unsigned char ret;  
93     unsigned short a = 0x0100;  
  
94     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/  
  
95     sink = ret;  
96 }  
97
```

01.w_Defects/data_lost.c:106

Level Medium**Status** Not processed

```
103 {  
104     unsigned short ret;  
105     unsigned int a = 0x00010000;
```

106 ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

```
107     sink = ret;  
108 }  
109
```

Trace

vflag ==1

01.w_Defects/data_lost.c:267

```
264 extern volatile int vflag;  
265 void data_lost_main ()  
266 {
```

267 if (vflag ==1 || vflag ==888)

```
268     {  
269         data_lost_001();  
270     }
```

ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

01.w_Defects/data_lost.c:106

```
103 {  
104     unsigned short ret;  
105     unsigned int a = 0x00010000;
```

```
106 ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision  
lost because of cast*/
```

```
107     sink = ret;  
108 }  
109
```

01.w_Defects/data_lost.c:118

Level Medium

Status Not processed

```
115 {  
116     unsigned short ret;  
117     unsigned long a = 0x00010000;
```

```
118 ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost  
because of cast*/
```

```
119     sink = ret;  
120 }  
121
```

Trace

vflag ==1

01.w_Defects/data_lost.c:267

```
264 extern volatile int vflag;  
265 void data_lost_main ()  
266 {
```

```
267     if (vflag ==1 || vflag ==888)
```

```
268     {  
269         data_lost_001();  
270     }
```

```
ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/
```

01.w_Defects/data_lost.c:118

```
115 {  
116     unsigned short ret;  
117     unsigned long a = 0x00010000;  
  
118     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/  
  
119     sink = ret;  
120 }  
121
```

01.w_Defects/data_lost.c:209

Level Medium

Status Not processed

```
206 void data_lost_016_func_001 (int a)  
207 {  
208     short ret;
```

```
209     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/
```

```
210     sink = ret;  
211 }  
212
```

Trace

```
vflag ==1
```

01.w_Defects/data_lost.c:267

```
264 extern volatile int vflag;
265 void data_lost_main ()
266 {
```

```
267 if (vflag ==1 || vflag ==888)
```

```
268 {
269     data_lost_001();
270 }
```

ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

01.w_Defects/data_lost.c:209

```
206 void data_lost_016_func_001 (int a)
207 {
208     short ret;
```

```
209     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/
```

```
210     sink = ret;
211 }
212
```

01.w_Defects/data_lost.c:240

Level Medium

Status Not processed

```
237 int a = 0x8000;
238 int a1;
239 a1 = a;
```

```
240 ret = a1; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost
```

because of cast*/

```
241     sink = ret;  
242 }  
243
```

Trace

vflag ==1

01.w_Defects/data_lost.c:267

```
264 extern volatile int vflag;  
265 void data_lost_main ()  
266 {  
  
267 if (vflag ==1 || vflag ==888)  
  
268 {  
269     data_lost_001();  
270 }
```

ret = a1; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

01.w_Defects/data_lost.c:240

```
237 int a = 0x8000;  
238 int a1;  
239 a1 = a;
```

240 ret = a1; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

```
241     sink = ret;  
242 }  
243
```

01.w_Defects/data_lost.c:256

Level Medium**Status** Not processed

```
253 int a2;  
254 a1 = a;  
255 a2 = a1;
```

256 ret = a2; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

```
257     sink = ret;  
258 }  
259
```

Trace

vflag ==1

01.w_Defects/data_lost.c:267

```
264 extern volatile int vflag;  
265 void data_lost_main ()  
266 {  
  
267     if (vflag ==1 || vflag ==888)  
  
268     {  
269         data_lost_001();  
270     }
```

ret = a2; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/

01.w_Defects/data_lost.c:256

```
253 int a2;  
254 a1 = a;  
255 a2 = a1;
```

```
256 ret = a2; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost because of cast*/
```

```
257     sink = ret;
258 }
259
```

01.w_Defects/data_overflow.c:62

Level Medium

Status Not processed

```
59 long max = 0x7fffffff;
60 long ret;
61 ret = max + 1; /*Tool should detect this line as error*/ /*ERROR:Data Overflow*/
```

```
62     sink = ret;
```

```
63 }
64
65 /*
```

Trace

```
vflag ==1
```

01.w_Defects/data_overflow.c:361

```
358 extern volatile int vflag;
359 void data_overflow_main ()
360 {
```

```
361 if (vflag ==1 || vflag ==888)
```

```
362 {
363     data_overflow_001();
364 }
```

```
sink = ret
```

01.w_Defects/data_overflow.c:62

```
59 long max = 0x7fffffff;
60 long ret;
61 ret = max + 1; /*Tool should detect this line as error*/ /*ERROR:Data
Overflow*/
62 sink = ret;
63 }
64
65 /*
```

01.w_Defects/data_overflow.c:110

Level Medium

Status Not processed

```
107 unsigned long max = 0xffffffff;
108 unsigned long ret;
109 ret = max + 1; /*Tool should detect this line as error*/ /*ERROR:Data Overflow*/
110 sink = ret;
111 }
112
113 /*
```

Trace

```
vflag ==1
```

01.w_Defects/data_overflow.c:361

```
358 extern volatile int vflag;
359 void data_overflow_main ()
360 {
```

```
361 if (vflag ==1 || vflag ==888)
```

```
362 {  
363     data_overflow_001();  
364 }
```

sink = ret

01.w_Defects/data_overflow.c:110

```
107 unsigned long max = 0xffffffff;  
108 unsigned long ret;  
109 ret = max + 1; /*Tool should detect this line as error*/ /*ERROR:Data  
Overflow*/
```

```
110     sink = ret;
```

```
111 }  
112  
113 /*
```

01.w_Defects/data_underflow.c:34

Level Medium

Status Not processed

```
31 unsigned int min = 0;  
32 unsigned int ret;  
33 ret = min - 1; /*Tool should detect this line as error*/ /*ERROR:Data Underflow*/
```

```
34     sink = ret;
```

```
35 }  
36  
37 /*
```

Trace

```
vflag ==1
```

01.w_Defects/data_underflow.c:182

```
179 extern volatile int vflag;
180 void data_underflow_main ()
181 {

182 if (vflag ==1 || vflag ==888)

183 {
184     data_underflow_001();
185 }
```

```
sink = ret
```

01.w_Defects/data_underflow.c:34

```
31 unsigned int min = 0;
32 unsigned int ret;
33 ret = min - 1; /*Tool should detect this line as error*/ /*ERROR:Data
Underflow*/

34 sink = ret;

35 }
36
37 /*
```

01.w_Defects/sign_conv.c:23

Level Medium

Status Not processed

```
20 {
21     unsigned char a = 0xff;
22     char ret;
```

```
23     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */
```

```
24     sink = ret;
25 }
26
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {
273     if (vflag == 1 || vflag == 888)
274     {
275         sign_conv_001();
276     }
```

ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */

01.w_Defects/sign_conv.c:23

```
20 {
21     unsigned char a = 0xff;
22     char ret;
23     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */
24     sink = ret;
25 }
26
```

01.w_Defects/sign_conv.c:35

Level Medium

Status Not processed

```
32 {  
33     unsigned short a = 0xffff;  
34     short ret;  
  
35     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
36     sink = ret;  
37 }  
38
```

Trace

```
vflag == 1
```

```
01.w_Defects/sign_conv.c:273
```

```
270 extern volatile int vflag;  
271 void sign_conv_main ()  
272 {  
  
273     if (vflag == 1 || vflag == 888)  
  
274     {  
275         sign_conv_001();  
276     }
```

```
ret = a; /*Tool should detect this line as  
error*/ /*Integer sign lost because of  
unsigned cast */
```

```
01.w_Defects/sign_conv.c:35
```

```
32 {  
33     unsigned short a = 0xffff;  
34     short ret;  
  
35     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
36     sink = ret;
```

```
37 }  
38
```

01.w_Defects/sign_conv.c:47

Level Medium

Status Not processed

```
44 {  
45     unsigned int a = 0xffffffff;  
46     int ret;  
  
47     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
48     sink = ret;  
49 }  
50
```

Trace

```
vflag == 1
```

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;  
271 void sign_conv_main ()  
272 {  
  
273     if (vflag == 1 || vflag == 888)  
  
274     {  
275         sign_conv_001();  
276     }
```

```
ret = a; /*Tool should detect this line as  
error*/ /*Integer sign lost because of  
unsigned cast */
```

01.w_Defects/sign_conv.c:47

```
44 {  
45     unsigned int a = 0xffffffff;  
46     int ret;  
  
47     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
48     sink = ret;  
49 }  
50
```

01.w_Defects/sign_conv.c:60

Level Medium

Status Not processed

```
57     unsigned long a = 0xffffffff;  
58     long ret;  
59     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
60     sink = ret;  
  
61 }  
62  
63 /*
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {
273     if (vflag == 1 || vflag == 888)
274     {
275         sign_conv_001();
276     }
}
```

sink = ret

01.w_Defects/sign_conv.c:60

```
57     unsigned long a = 0xffffffff;
58     long ret;
59     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */
60     sink = ret;
61 }
62
63 /*
```

01.w_Defects/sign_conv.c:71

Level Medium

Status Not processed

```
68 {
69     char a = -1;
70     unsigned char ret;
```

```
71     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */
```

```
72     sink = ret;
73 }
74
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {
273     if (vflag == 1 || vflag == 888)
274     {
275         sign_conv_001();
276     }
```

ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */

01.w_Defects/sign_conv.c:71

```
68 {
69     char a = -1;
70     unsigned char ret;
71     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */
72     sink = ret;
73 }
74
```

01.w_Defects/sign_conv.c:83

Level Medium

Status Not processed

```
80 {  
81   short a = -1;  
82   unsigned short ret;  
  
83   ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
84   sink = ret;  
85 }  
86
```

Trace

```
vflag == 1
```

```
01.w_Defects/sign_conv.c:273
```

```
270 extern volatile int vflag;  
271 void sign_conv_main ()  
272 {  
  
273   if (vflag == 1 || vflag == 888)  
  
274   {  
275     sign_conv_001();  
276   }
```

```
ret = a; /*Tool should detect this line as  
error*/ /*Integer sign lost because of  
unsigned cast */
```

```
01.w_Defects/sign_conv.c:83
```

```
80 {  
81   short a = -1;  
82   unsigned short ret;  
  
83   ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
84   sink = ret;
```

```
85 }  
86
```

01.w_Defects/sign_conv.c:95

Level Medium**Status** Not processed

```
92 {  
93     int a = -1;  
94     unsigned int ret;  
  
95     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
96     sink = ret;  
97 }  
98
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;  
271 void sign_conv_main ()  
272 {  
  
273     if (vflag == 1 || vflag == 888)  
  
274     {  
275         sign_conv_001();  
276     }
```

```
ret = a; /*Tool should detect this line as  
error*/ /*Integer sign lost because of  
unsigned cast */
```

01.w_Defects/sign_conv.c:95

```
92 {  
93     int a = -1;  
94     unsigned int ret;  
  
95     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
96     sink = ret;  
97 }  
98
```

01.w_Defects/sign_conv.c:96

Level Medium

Status Not processed

```
93     int a = -1;  
94     unsigned int ret;  
95     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
96     sink = ret;  
  
97 }  
98  
99 /*
```

Trace

```
vflag == 1
```

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {
273     if (vflag == 1 || vflag == 888)
274     {
275         sign_conv_001();
276     }
}
```

```
sink = ret
```

01.w_Defects/sign_conv.c:96

```
93     int a = -1;
94     unsigned int ret;
95     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */
96     sink = ret;
97 }
98
99 /*
```

01.w_Defects/sign_conv.c:107

Level Medium

Status Not processed

```
104 {
105     long a = -1;
106     unsigned long ret;
```

```
107     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */
```

```
108     sink = ret;
109 }
110
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {
273     if (vflag == 1 || vflag == 888)
274     {
275         sign_conv_001();
276     }
```

ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */

01.w_Defects/sign_conv.c:107

```
104 {
105     long a = -1;
106     unsigned long ret;
107     ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */
108     sink = ret;
109 }
110
```

01.w_Defects/sign_conv.c:108

Level Medium

Status Not processed

```
105 long a = -1;
106 unsigned long ret;
107 ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */

108     sink = ret;

109 }
110
111 /*
```

Trace

```
vflag == 1
```

```
01.w_Defects/sign_conv.c:273
```

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {

273 if (vflag == 1 || vflag == 888)

274 {
275     sign_conv_001();
276 }
```

```
sink = ret
```

```
01.w_Defects/sign_conv.c:108
```

```
105 long a = -1;
106 unsigned long ret;
107 ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */

108     sink = ret;

109 }
110
111 /*
```

01.w_Defects/sign_conv.c:151

Level Medium**Status** Not processed

```
148 {  
149     unsigned int ret;  
150     ret = -1; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
151     sink = ret;  
  
152 }  
153  
154 /*
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;  
271 void sign_conv_main ()  
272 {  
  
273     if (vflag == 1 || vflag == 888)  
  
274     {  
275         sign_conv_001();  
276     }
```

```
sink = ret
```

01.w_Defects/sign_conv.c:151

```
148 {  
149     unsigned int ret;  
150     ret = -1; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
151     sink = ret;  
  
152 }  
153  
154 /*
```

01.w_Defects/sign_conv.c:180

Level Medium

Status Not processed

```
177     int a = -1;  
178     unsigned int ret;  
179     ret = (5 * a) + 4; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */  
  
180     sink = ret;  
  
181 }  
182  
183 /*
```

Trace

```
vflag == 1
```

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;  
271 void sign_conv_main ()  
272 {
```

```
273 if (vflag == 1 || vflag ==888)
```

```
274 {  
275     sign_conv_001();  
276 }
```

```
sink = ret
```

01.w_Defects/sign_conv.c:180

```
177 int a = -1;  
178 unsigned int ret;  
179 ret = (5 * a) + 4; /*Tool should detect this line as error*/ /*Integer sign lost  
because of unsigned cast */
```

```
180     sink = ret;
```

```
181 }  
182  
183 /*
```

01.w_Defects/sign_conv.c:192

Level Medium

Status Not processed

```
189 int a = 2;  
190 unsigned int ret;  
191 ret = (a * a) - 5; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */
```

```
192     sink = ret;
```

```
193 }  
194  
195 /*
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {
273     if (vflag == 1 || vflag == 888)
274     {
275         sign_conv_001();
276     }
}
```

sink = ret

01.w_Defects/sign_conv.c:192

```
189 int a = 2;
190 unsigned int ret;
191 ret = (a * a) - 5; /*Tool should detect this line as error*/ /*Integer sign lost
because of unsigned cast */
192     sink = ret;
193 }
194 /*
195 */


```

01.w_Defects/sign_conv.c:217

Level Medium

Status Not processed

```
214 unsigned int sign_conv_016_gbl_ret;
215 void sign_conv_016_func_001 (int a)
216 {
```

```
217     sign_conv_016_gbl_ret = a; /*Tool should detect this line as error*/ /*Integer sign
lost because of unsigned cast */
```

```
218 }  
219  
220 void sign_conv_016 ()
```

Trace

```
vflag == 1
```

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;  
271 void sign_conv_main ()  
272 {  
  
273 if (vflag == 1 || vflag ==888)  
  
274 {  
275     sign_conv_001();  
276 }
```

sign_conv_016_gbl_ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */

01.w_Defects/sign_conv.c:217

```
214 unsigned int sign_conv_016_gbl_ret;  
215 void sign_conv_016_func_001 (int a)  
216 {
```

217 sign_conv_016_gbl_ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */

```
218 }  
219  
220 void sign_conv_016 ()
```

01.w_Defects/sign_conv.c:246

Level Medium

Status Not processed

```
243 int a1;
244 unsigned int ret;
245 a1 = a;

246 ret = a1; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */

247     sink = ret;
248 }
249
```

Trace

```
vflag == 1
```

```
01.w_Defects/sign_conv.c:273
```

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {

273 if (vflag == 1 || vflag == 888)

274 {
275     sign_conv_001();
276 }
```

```
ret = a1; /*Tool should detect this line as
error*/ /*Integer sign lost because of
unsigned cast */
```

```
01.w_Defects/sign_conv.c:246
```

```
243 int a1;
244 unsigned int ret;
245 a1 = a;

246 ret = a1; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */

247     sink = ret;
```

```
248 }  
249
```

01.w_Defects/sign_conv.c:247

Level Medium

Status Not processed

```
244 unsigned int ret;  
245 a1 = a;  
246 ret = a1; /*Tool should detect this line as error*/ /*Integer sign lost because of  
unsigned cast */
```

247 sink = ret;

```
248 }  
249  
250 /*
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;  
271 void sign_conv_main ()  
272 {  
  
273 if (vflag == 1 || vflag == 888)  
  
274 {  
275     sign_conv_001();  
276 }
```

sink = ret

01.w_Defects/sign_conv.c:247

```
244 unsigned int ret;
245 a1 = a;
246 ret = a1; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */

247     sink = ret;

248 }
249
250 /*
```

01.w_Defects/sign_conv.c:262

Level Medium

Status Not processed

```
259 unsigned int ret;
260 a1 = a;
261 a2 = a1;
```

262 ret = a2; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */

```
263     sink = ret;
264 }
265
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {
```

```
273 if (vflag == 1 || vflag ==888)
```

```
274 {  
275     sign_conv_001();  
276 }
```

ret = a2; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */

01.w_Defects/sign_conv.c:262

```
259 unsigned int ret;  
260 a1 = a;  
261 a2 = a1;
```

262 ret = a2; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */

```
263     sink = ret;  
264 }  
265
```

01.w_Defects/sign_conv.c:263

Level Medium

Status Not processed

```
260 a1 = a;  
261 a2 = a1;  
262 ret = a2; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned cast */
```

```
263     sink = ret;
```

```
264 }  
265  
266 /*
```

Trace

vflag == 1

01.w_Defects/sign_conv.c:273

```
270 extern volatile int vflag;
271 void sign_conv_main ()
272 {
273     if (vflag == 1 || vflag ==888)
274     {
275         sign_conv_001();
276     }
```

sink = ret

01.w_Defects/sign_conv.c:263

```
260     a1 = a;
261     a2 = a1;
262     ret = a2; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */
263     sink = ret;
264 }
265
266 /*
```

02.wo_Defects/data_overflow.c:99

Level Medium

Status Not processed

```
96     unsigned int max = 0xffffffff;
97     unsigned int ret;
98     ret = max + 1; /*Tool should not detect this line as error*/ /*No ERROR:Data
Overflow*/
```

```
99     sink = ret;
```

```
100 }
```

```
101
```

```
102 /*
```

Trace

```
vflag ==1
```

```
02.wo_Defects/data_overflow.c:363
```

```
360 extern volatile int vflag;
361 void data_overflow_main ()
362 {
```

```
363 if (vflag ==1 || vflag ==888)
```

```
364 {
365     data_overflow_001();
366 }
```

```
sink = ret
```

```
02.wo_Defects/data_overflow.c:99
```

```
96     unsigned int max = 0xffffffff;
97     unsigned int ret;
98     ret = max + 1; /*Tool should not detect this line as error*/ /*No ERROR:Data
Overflow*/
```

```
99     sink = ret;
```

```
100 }
```

```
101
```

```
102 /*
```

```
02.wo_Defects/data_overflow.c:111
```

Level Medium

Status Not processed

```
108 unsigned long max = 0xffffffff;
109 unsigned long ret;
110 ret = max + 1; /*Tool should not detect this line as error*/ /*No ERROR:Data
Overflow*/
111     sink = ret;
112 }
113
114 /*
```

Trace

```
vflag ==1
```

```
02.wo_Defects/data_overflow.c:363
```

```
360 extern volatile int vflag;
361 void data_overflow_main ()
362 {
```

```
363     if (vflag ==1 || vflag ==888)
```

```
364     {
365         data_overflow_001();
366     }
```

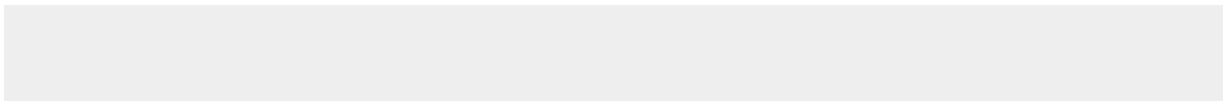
```
sink = ret
```

```
02.wo_Defects/data_overflow.c:111
```

```
108 unsigned long max = 0xffffffff;
109 unsigned long ret;
110 ret = max + 1; /*Tool should not detect this line as error*/ /*No ERROR:Data
Overflow*/
```

```
111     sink = ret;
```

```
112 }
113
114 /*
```



Malloc overflow (C/C++)

Description

Overflow may occur when calculating the size of the allocated memory. The malloc function allocates a block of memory with the specified argument size and returns a pointer to the beginning of a block. Memory allocated using malloc() function not only includes the user block but also data used to manage the heap (the size of block, pointer to other blocks), thus, a heap overflow may lead to overwriting these control data. This may lead to incorrect behavior, crash of the application or degrading system response time.

Example

In the following example, overflow may occur when calculating the allocated memory size:

```
void * f1(int n)
{
    return malloc(n * sizeof(int));
}
```

Recommendations

- Check the values of arguments of function that allocate memory and do not allow overflow.

Links

1. OWASP: Heap overflow
2. CWE-122: Heap-based Buffer Overflow

Vulnerability Entries

01.w_Defects/free_null_pointer.c:102

Level Medium

Status Not processed

```
99 {  
100 free_null_pointer_005_gbl_ptr=NULL;  
101 if(a != INDEX)  
  
102     free_null_pointer_005_gbl_ptr= malloc(sizeof(char) * (len+1));  
  
103 }  
104  
105 void free_null_pointer_005 ()
```

Trace

sizeof(char) * (len+1)

01.w_Defects/free_null_pointer.c:102

```
99 {  
100 free_null_pointer_005_gbl_ptr=NULL;  
101 if(a != INDEX)  
  
102     free_null_pointer_005_gbl_ptr= malloc(sizeof(char) * (len+1));  
  
103 }  
104  
105 void free_null_pointer_005 ()
```

sizeof(char) * (len+1)

01.w_Defects/free_null_pointer.c:102

```
99 {  
100 free_null_pointer_005_gbl_ptr=NULL;  
101 if(a != INDEX)  
  
102     free_null_pointer_005_gbl_ptr= malloc(sizeof(char) * (len+1));
```

```
103 }  
104  
105 void free_null_pointer_005 ()
```

01.w_Defects/free_null_pointer.c:231

Level Medium

Status Not processed

```
228 char * p = NULL;  
229 if(min <= min_buffer && max <= max_buffer)  
230 {  
  
231     p = malloc(sizeof(char) * (len+1));  
  
232     *stringPtr = p;  
233 }  
234 }
```

Trace

sizeof(char) * (len+1)

01.w_Defects/free_null_pointer.c:231

```
228 char * p = NULL;  
229 if(min <= min_buffer && max <= max_buffer)  
230 {  
  
231     p = malloc(sizeof(char) * (len+1));  
  
232     *stringPtr = p;  
233 }  
234 }
```

```
sizeof(char) * (len+1)
```

01.w_Defects/free_null_pointer.c:231

```
228 char * p = NULL;
229 if(min <= min_buffer && max <= max_buffer)
230 {
231     p = malloc(sizeof(char) * (len+1));
232     *stringPtr = p;
233 }
234 }
```

01.w_Defects/invalid_memory_access.c:280

Level Medium

Status Not processed

```
277 */
278 void invalid_memory_access_func_010 (int len ,int **Ptr)
279 {

280 int * p = malloc(sizeof(int) * len);

281 *Ptr = p;
282 }
283
```

Trace

```
sizeof(int) * len
```

01.w_Defects/invalid_memory_access.c:280

```
277 */
278 void invalid_memory_access_func_010 (int len ,int **Ptr)
279 {

280 int * p = malloc(sizeof(int) * len);
```

```
281 *Ptr = p;  
282 }  
283
```

sizeof(int) * len

01.w_Defects/invalid_memory_access.c:280

```
277 */  
278 void invalid_memory_access_func_010 (int len ,int **Ptr)  
279 {  
  
280 int * p = malloc(sizeof(int) * len);  
  
281 *Ptr = p;  
282 }  
283
```

01.w_Defects/memory_allocation_failure.c:258

Level Medium

Status Not processed

```
255 int i;  
256 for(i=0;i<max_buffer;i++)  
257 {  
  
258 buffer = (char*) malloc(max_buffer * sizeof(char));/*Tool should detect this line as  
error*/ /*ERROR:Memory allocation failure */  
  
259 break;  
260 }  
261 if(buffer!=NULL)
```

Trace

max_buffer * sizeof(char)

01.w_Defects/memory_allocation_failure.c:258

```
255 int i;
256 for(i=0;i<max_buffer;i++)
257 {
258   buffer = (char*) malloc(max_buffer * sizeof(char));/*Tool should detect this
line as error*/ /*ERROR:Memory allocation failure */
259   break;
260 }
261 if(buffer!=NULL)
```

max_buffer * sizeof(char)

01.w_Defects/memory_allocation_failure.c:258

```
255 int i;
256 for(i=0;i<max_buffer;i++)
257 {
258   buffer = (char*) malloc(max_buffer * sizeof(char));/*Tool should detect this
line as error*/ /*ERROR:Memory allocation failure */
259   break;
260 }
261 if(buffer!=NULL)
```

01.w_Defects/memory_leak.c:64

Level Medium

Status Not processed

```
61 */
62 void memory_leak_003_func_001 (int len,char **stringPtr)
63 {
64   char * p = malloc(sizeof(char) * (len+1));
```

```
65     *stringPtr = p;
66 }
67
```

Trace

sizeof(char) * (len+1)

01.w_Defects/memory_leak.c:64

```
61 */
62 void memory_leak_003_func_001 (int len,char **stringPtr)
63 {

64     char * p = malloc(sizeof(char) * (len+1));

65     *stringPtr = p;
66 }
67
```

sizeof(char) * (len+1)

01.w_Defects/memory_leak.c:64

```
61 */
62 void memory_leak_003_func_001 (int len,char **stringPtr)
63 {

64     char * p = malloc(sizeof(char) * (len+1));

65     *stringPtr = p;
66 }
67
```

01.w_Defects/memory_leak.c:392

Level Medium

Status Not processed

```
389 */
390 char * memory_leak_0015_func_001 (int len)
391 {

392     char *stringPtr = malloc(sizeof(char) * (len+1));

393     return stringPtr;
394 }
395
```

Trace

```
    sizeof(char) * (len+1)
```

01.w_Defects/memory_leak.c:392

```
389 */
390 char * memory_leak_0015_func_001 (int len)
391 {

392     char *stringPtr = malloc(sizeof(char) * (len+1));

393     return stringPtr;
394 }
395
```

```
    sizeof(char) * (len+1)
```

01.w_Defects/memory_leak.c:392

```
389 */
390 char * memory_leak_0015_func_001 (int len)
391 {

392     char *stringPtr = malloc(sizeof(char) * (len+1));

393     return stringPtr;
394 }
395
```

01.w_Defects/memory_leak.c:417

Level Medium**Status** Not processed

```
414 {  
415 memory_leak_0016_gbl_ptr=NULL;  
416 if(a == INDEX)  
  
417         memory_leak_0016_gbl_ptr= malloc(sizeof(char) * (len+1));/*Tool should  
detect this line as error*/ /*ERROR:Memory Leakage */  
  
418 }  
419  
420 void memory_leak_0016 ()
```

Trace

sizeof(char) * (len+1)

01.w_Defects/memory_leak.c:417

```
414 {  
415 memory_leak_0016_gbl_ptr=NULL;  
416 if(a == INDEX)  
  
417         memory_leak_0016_gbl_ptr= malloc(sizeof(char) * (len+1));/*Tool  
should detect this line as error*/ /*ERROR:Memory Leakage */  
  
418 }  
419  
420 void memory_leak_0016 ()
```

sizeof(char) * (len+1)

01.w_Defects/memory_leak.c:417

```
414 {  
415 memory_leak_0016_gbl_ptr=NULL;  
416 if(a == INDEX)  
  
417         memory_leak_0016_gbl_ptr= malloc(sizeof(char) * (len+1));/*Tool
```

```
should detect this line as error*/ /*ERROR:Memory Leakage */
```

```
418 }
419
420 void memory_leak_0016 ()
```

01.w_Defects/null_pointer.c:231

Level Medium

Status Not processed

```
228 {
229   null_pointer_015_gbl_ptr=NULL;
230   if(a != INDEX)

231       null_pointer_015_gbl_ptr= malloc(sizeof(char) * (len+1));

232 }
233
234 void null_pointer_015 ()
```

Trace

sizeof(char) * (len+1)

01.w_Defects/null_pointer.c:231

```
228 {
229   null_pointer_015_gbl_ptr=NULL;
230   if(a != INDEX)

231       null_pointer_015_gbl_ptr= malloc(sizeof(char) * (len+1));

232 }
233
234 void null_pointer_015 ()
```

```
sizeof(char) * (len+1)
```

01.w_Defects/null_pointer.c:231

```
228 {  
229     null_pointer_015_gbl_ptr=NULL;  
230     if(a != INDEX)  
  
231         null_pointer_015_gbl_ptr= malloc(sizeof(char) * (len+1));  
  
232 }  
233  
234 void null_pointer_015 ()
```

01.w_Defects/wrong_arguments_func_pointer.c:371

Level Medium

Status Not processed

```
368 */  
369 void wrong_arguments_func_pointer_013_func_001 (int len,char **stringPtr)  
370 {  
  
371     char * p = malloc(sizeof(char) * (len+1));  
  
372     *stringPtr = p;  
373 }  
374
```

Trace

```
sizeof(char) * (len+1)
```

01.w_Defects/wrong_arguments_func_pointer.c:371

```
368 */  
369 void wrong_arguments_func_pointer_013_func_001 (int len,char **stringPtr)  
370 {  
  
371     char * p = malloc(sizeof(char) * (len+1));
```

```
372 *stringPtr = p;
373 }
374
```

```
sizeof(char) * (len+1)
```

01.w_Defects/wrong_arguments_func_pointer.c:371

```
368 */
369 void wrong_arguments_func_pointer_013_func_001 (int len,char **stringPtr)
370 {

371     char * p = malloc(sizeof(char) * (len+1));

372     *stringPtr = p;
373 }
374
```

02.wo_Defects/free_null_pointer.c:112

Level Medium

Status Not processed

```
109 {
110     free_null_pointer_005_gbl_ptr=NULL;
111     if(a == INDEX)

112         free_null_pointer_005_gbl_ptr= malloc(sizeof(char) * (len+1));

113 }
114
115 void free_null_pointer_005 ()
```

Trace

```
sizeof(char) * (len+1)
```

02.wo_Defects/free_null_pointer.c:112

```
109 {  
110   free_null_pointer_005_gbl_ptr=NULL;  
111   if(a == INDEX)  
  
112       free_null_pointer_005_gbl_ptr= malloc(sizeof(char) * (len+1));  
  
113 }  
114  
115 void free_null_pointer_005 ()
```

```
sizeof(char) * (len+1)
```

02.wo_Defects/free_null_pointer.c:112

```
109 {  
110   free_null_pointer_005_gbl_ptr=NULL;  
111   if(a == INDEX)  
  
112       free_null_pointer_005_gbl_ptr= malloc(sizeof(char) * (len+1));  
  
113 }  
114  
115 void free_null_pointer_005 ()
```

02.wo_Defects/free_null_pointer.c:239

Level Medium

Status Not processed

```
236 char * p = NULL;  
237 if(min <= min_buffer && max <= max_buffer)  
238 {  
  
239     p = malloc(sizeof(char) * (len+1));  
  
240     *stringPtr = p;
```

```
241 }
242 }
```

Trace

```
sizeof(char) * (len+1)
```

02.wo_Defects/free_null_pointer.c:239

```
236 char * p = NULL;
237 if(min <= min_buffer && max <= max_buffer)
238 {
239     p = malloc(sizeof(char) * (len+1));
240     *stringPtr = p;
241 }
242 }
```

```
sizeof(char) * (len+1)
```

02.wo_Defects/free_null_pointer.c:239

```
236 char * p = NULL;
237 if(min <= min_buffer && max <= max_buffer)
238 {
239     p = malloc(sizeof(char) * (len+1));
240     *stringPtr = p;
241 }
242 }
```

02.wo_Defects/invalid_memory_access.c:289

Level Medium

Status Not processed

```
286 */
```

```
287 void invalid_memory_access_func_010 (int len ,int **Ptr)
288 {
289   int * p = malloc(sizeof(int) * len);
290   *Ptr = p;
291 }
292
```

Trace

```
sizeof(int) * len
```

02.wo_Defects/invalid_memory_access.c:289

```
286 */
287 void invalid_memory_access_func_010 (int len ,int **Ptr)
288 {
289   int * p = malloc(sizeof(int) * len);
290   *Ptr = p;
291 }
292
```

```
sizeof(int) * len
```

02.wo_Defects/invalid_memory_access.c:289

```
286 */
287 void invalid_memory_access_func_010 (int len ,int **Ptr)
288 {
289   int * p = malloc(sizeof(int) * len);
290   *Ptr = p;
291 }
292
```

02.wo_Defects/memory_allocation_failure.c:267

Level Medium**Status** Not processed

```
264 int i;
265 for(i=0;i<max_buffer;i++)
266 {
267     buffer = (char*) malloc(max_buffer * sizeof(char)); /*Tool should not detect this line
as error*/ /*No ERROR:Memory allocation failure */
268     break;
269 }
270 if(buffer!=NULL)
```

Trace

max_buffer * sizeof(char)

02.wo_Defects/memory_allocation_failure.c:267

```
264 int i;
265 for(i=0;i<max_buffer;i++)
266 {
267     buffer = (char*) malloc(max_buffer * sizeof(char)); /*Tool should not detect
this line as error*/ /*No ERROR:Memory allocation failure */
268     break;
269 }
270 if(buffer!=NULL)
```

max_buffer * sizeof(char)

02.wo_Defects/memory_allocation_failure.c:267

```
264 int i;
265 for(i=0;i<max_buffer;i++)
266 {
267     buffer = (char*) malloc(max_buffer * sizeof(char)); /*Tool should not detect
```

```
this line as error*/ /*No ERROR:Memory allocation failure */
```

```
268     break;  
269 }  
270 if(buffer!=NULL)
```

02.wo_Defects/memory_leak.c:67

Level Medium

Status Not processed

```
64 */  
65 void memory_leak_003_func_001 (int len,char **stringPtr)  
66 {  
  
67     char * p = malloc(sizeof(char) * (len+1));  
  
68     *stringPtr = p;  
69 }  
70
```

Trace

sizeof(char) * (len+1)

02.wo_Defects/memory_leak.c:67

```
64 */  
65 void memory_leak_003_func_001 (int len,char **stringPtr)  
66 {  
  
67     char * p = malloc(sizeof(char) * (len+1));  
  
68     *stringPtr = p;  
69 }  
70
```

```
sizeof(char) * (len+1)
```

02.wo_Defects/memory_leak.c:67

```
64 */
65 void memory_leak_003_func_001 (int len,char **stringPtr)
66 {

67     char * p = malloc(sizeof(char) * (len+1));

68     *stringPtr = p;
69 }
70
```

02.wo_Defects/memory_leak.c:398

Level Medium

Status Not processed

```
395 */
396 char * memory_leak_0015_func_001 (int len)
397 {

398     char *stringPtr = malloc(sizeof(char) * (len+1));

399     return stringPtr;
400 }
401
```

Trace

```
sizeof(char) * (len+1)
```

02.wo_Defects/memory_leak.c:398

```
395 */
396 char * memory_leak_0015_func_001 (int len)
397 {

398     char *stringPtr = malloc(sizeof(char) * (len+1));
```

```
399 return stringPtr;
400 }
401
```

```
sizeof(char) * (len+1)
```

02.wo_Defects/memory_leak.c:398

```
395 */
396 char * memory_leak_0015_func_001 (int len)
397 {
```

```
398     char *stringPtr = malloc(sizeof(char) * (len+1));
```

```
399     return stringPtr;
400 }
401
```

02.wo_Defects/memory_leak.c:424

Level Medium

Status Not processed

```
421 {
422     memory_leak_0016_gbl_ptr=NULL;
423     if(a == INDEX)
```

```
424         memory_leak_0016_gbl_ptr= malloc(sizeof(char) * (len+1)); /*Tool should
not detect this line as error*/ /*No ERROR:Memory Leakage */
```

```
425 }
426
427 void memory_leak_0016 ()
```

Trace

```
sizeof(char) * (len+1)
```

02.wo_Defects/memory_leak.c:424

```
421 {  
422     memory_leak_0016_gbl_ptr=NULL;  
423     if(a == INDEX)  
  
424         memory_leak_0016_gbl_ptr= malloc(sizeof(char) * (len+1)); /*Tool  
should not detect this line as error*/ /*No ERROR:Memory Leakage */  
  
425 }  
426  
427 void memory_leak_0016 ()
```

```
sizeof(char) * (len+1)
```

02.wo_Defects/memory_leak.c:424

```
421 {  
422     memory_leak_0016_gbl_ptr=NULL;  
423     if(a == INDEX)  
  
424         memory_leak_0016_gbl_ptr= malloc(sizeof(char) * (len+1)); /*Tool  
should not detect this line as error*/ /*No ERROR:Memory Leakage */  
  
425 }  
426  
427 void memory_leak_0016 ()
```

02.wo_Defects/null_pointer.c:252

Level Medium

Status Not processed

```
249 {  
250     null_pointer_015_gbl_ptr=NULL;  
251     if(a == INDEX)
```

```
252         null_pointer_015_gbl_ptr= malloc(sizeof(char) * (len+1));
```

```
253 }  
254  
255 void null_pointer_015 ()
```

Trace

```
    sizeof(char) * (len+1)
```

02.wo_Defects/null_pointer.c:252

```
249 {  
250     null_pointer_015_gbl_ptr=NULL;  
251     if(a == INDEX)  
  
252         null_pointer_015_gbl_ptr= malloc(sizeof(char) * (len+1));  
  
253 }  
254  
255 void null_pointer_015 ()
```

```
    sizeof(char) * (len+1)
```

02.wo_Defects/null_pointer.c:252

```
249 {  
250     null_pointer_015_gbl_ptr=NULL;  
251     if(a == INDEX)  
  
252         null_pointer_015_gbl_ptr= malloc(sizeof(char) * (len+1));  
  
253 }  
254  
255 void null_pointer_015 ()
```

02.wo_Defects/wrong_arguments_func_pointer.c:370

Level Medium

Status Not processed

```
367 */  
368 void wrong_arguments_func_pointer_013_func_001 (int len,char **stringPtr)  
369 {  
  
370 char * p = malloc(sizeof(char) * (len+1));  
  
371 *stringPtr = p;  
372 }  
373
```

Trace

```
sizeof(char) * (len+1)
```

02.wo_Defects/wrong_arguments_func_pointer.c:370

```
367 */  
368 void wrong_arguments_func_pointer_013_func_001 (int len,char **stringPtr)  
369 {  
  
370 char * p = malloc(sizeof(char) * (len+1));  
  
371 *stringPtr = p;  
372 }  
373
```

```
sizeof(char) * (len+1)
```

02.wo_Defects/wrong_arguments_func_pointer.c:370

```
367 */  
368 void wrong_arguments_func_pointer_013_func_001 (int len,char **stringPtr)  
369 {  
  
370 char * p = malloc(sizeof(char) * (len+1));  
  
371 *stringPtr = p;  
372 }  
373
```

Memory leak (C/C++)

Description

The application allows incorrect work with memory. Some errors related to memory management: no release of allocated memory (memory leak), double memory allocation, the use of the wrong class to free memory.

C language involves working with memory at a relatively low level. The developer must allocate memory for the structures used and release it correctly after the end of the work. Otherwise, memory available for the application will decrease, eventually causing malfunction of the application.

Example

In the following example, the array is created and there is no corresponding free function:

```
char *p = malloc ( 10 );
```

```
...
```

```
// no free(p)
```

Recommendations

- Free the allocated memory correctly.
- Check the documentation of methods of work with memory.

Links

1. CWE-401: Improper Release of Memory Before Removing Last Reference ('Memory Leak')
2. Pointers and memory leaks in C

Vulnerability Entries

01.w_Defects/double_free.c:88

Level Medium**Status** Not processed

```
85
86 if(rand() % 3==0)
87 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
```

```
88 }
```

```
89
90 /*
91 * Types of defects: Double free
```

Trace

```
vflag == 1
```

01.w_Defects/double_free.c:233

```
230 extern volatile int vflag;
231 void double_free_main ()
232 {
233     if (vflag == 1 || vflag ==888)
234     {
235         double_free_001 ();
236     }
}
```

01.w_Defects/double_free.c:88

```
85
86 if(rand() % 3==0)
87 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
88 }
89
```

```
90 /*  
91 * Types of defects: Double free
```

01.w_Defects/func_pointer.c:117

Level Medium

Status Not processed

```
114 int j;  
115 char buf[][25]={"This is a String",  
116           "Second String"};
```

```
117 for(j = 0; j <= 1; j++)
```

```
118 {  
119 {  
120     char str;
```

Trace

```
vflag == 1
```

01.w_Defects/func_pointer.c:617

```
614 extern volatile int vflag;  
615 void func_pointer_main ()  
616 {
```

```
617 if (vflag == 1 || vflag ==888)
```

```
618 {  
619     func_pointer_001();  
620 }
```

```
for(j = 0; j <= 1; j++)
```

01.w_Defects/func_pointer.c:117

```
114 int j;
115 char buf[][25]={"This is a String",
116                 "Second String"};
```

```
117 for(j = 0; j <= 1; j++)
```

```
118 {
119 {
120     char str;
```

01.w_Defects/func_pointer.c:411

Level Medium

Status Not processed

```
408 int j;
409 const char buf[][25]={"This is a String",
410                         "Second String"};
```

```
411 for(j = 0; j <= 1; j++)
```

```
412 {
413     if (MAX ==10)
414 {
```

Trace

```
vflag == 1
```

01.w_Defects/func_pointer.c:617

```
614 extern volatile int vflag;
615 void func_pointer_main ()
616 {
```

```
617 if (vflag == 1 || vflag ==888)
```

```
618 {  
619     func_pointer_001();  
620 }
```

```
for(j = 0; j <= 1; j++)
```

01.w_Defects/func_pointer.c:411

```
408 int j;  
409 const char buf[][25]={"This is a String",  
410             "Second String"};
```

```
411 for(j = 0; j <= 1; j++)
```

```
412 {  
413     if (MAX ==10)  
414     {
```

01.w_Defects/func_pointer.c:608

Level Medium

Status Not processed

```
605 void (*fptr3)(func_pointer_015_s_001* st1);  
606 fptr3 = func_pointer_015_func_004;  
607 fptr3(st1);
```

```
608 }
```

```
609  
610 /*  
611 * Type of defect: bad function pointer casting
```

Trace

```
vflag == 1
```

01.w_Defects/func_pointer.c:617

```
614 extern volatile int vflag;
615 void func_pointer_main ()
616 {
617     if (vflag == 1 || vflag ==888)
618     {
619         func_pointer_001();
620     }
}
```

01.w_Defects/func_pointer.c:608

```
605 void (*fptr3)(func_pointer_015_s_001* st1);
606 fptr3 = func_pointer_015_func_004;
607 fptr3(st1);

608 }

609
610 /*
611 * Type of defect: bad function pointer casting
```

01.w_Defects/memory_allocation_failure.c:283

Level Medium

Status Not processed

```
280     do
281     {
282         buf = (char*) malloc(MAX_BUFFER * sizeof(char));/*Tool should detect this
line as error*/ /*ERROR:Memory allocation failure */
283         i++;
}
```

```
284 }while (i<MAX_VAL);
285 }
286
```

Trace

vflag == 1

01.w_Defects/memory_allocation_failure.c:724

```
721 extern volatile int vflag;
722 void memory_allocation_failure_main ()
723 {
724     if (vflag == 1 || vflag ==888)
725     {
726         memory_allocation_failure_001();
727     }
```

i++

01.w_Defects/memory_allocation_failure.c:283

```
280     do
281     {
282         buf = (char*) malloc(MAX_BUFFER * sizeof(char));/*Tool should
detect this line as error*/ /*ERROR:Memory allocation failure */
283         i++;
284     }while (i<MAX_VAL);
285 }
286
```

01.w_Defects/memory_leak.c:74

Level Medium

Status Not processed

```
71 char *str1;
72 memory_leak_003_func_001(strlen(str),&str1);/*Tool should detect this line as
error*/ /*ERROR:Memory Leakage */
73 strcpy(str1,str);

74 }

75
76 /*
77 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

Trace

```
vflag == 1
```

```
01.w_Defects/memory_leak.c:539
```

```
536 extern volatile int vflag;
537 void memory_leak_main ()
538 {

539 if (vflag == 1 || vflag ==888)

540 {
541     memory_leak_001();
542 }
```

```
}
```

```
01.w_Defects/memory_leak.c:74
```

```
71 char *str1;
72 memory_leak_003_func_001(strlen(str),&str1);/*Tool should detect this line as
error*/ /*ERROR:Memory Leakage */
73 strcpy(str1,str);

74 }

75
76 /*
77 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

01.w_Defects/memory_leak.c:120

Level Medium**Status** Not processed

```
117 }
118 if(flag < 0)
119 free(ptr);
```

```
120 }
```

```
121
122 /*
```

```
123 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

Trace

```
vflag == 1
```

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;
537 void memory_leak_main ()
538 {
539     if (vflag == 1 || vflag == 888)
540     {
541         memory_leak_001();
542     }
}
```

01.w_Defects/memory_leak.c:120

```
117 }
118 if(flag < 0)
119 free(ptr);
```

```
120 }
```

```
121
```

```
122 /*
```

```
123 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

01.w_Defects/memory_leak.c:151

Level Medium

Status Not processed

```
148 }
149 if(memory_leak_006_func_001(flag) !=0)
150 free(dptr);
```

151 }

```
152
153 /*
154 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

Trace

vflag == 1

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;
537 void memory_leak_main ()
538 {
```

539 if (vflag == 1 || vflag ==888)

```
540 {
541     memory_leak_001();
542 }
```

```
}
```

01.w_Defects/memory_leak.c:151

```
148 }
149 if(memory_leak_006_func_001(flag) !=0)
150 free(dptr);
```

```
151 }
```

```
152
153 /*
154 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

01.w_Defects/memory_leak.c:219

Level Medium

Status Not processed

```
216 *(p+1) = 1;
217 free(ptr);
218 }
```

```
219 }
```

```
220
221 /*
222 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

Trace

```
vflag == 1
```

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;
537 void memory_leak_main ()
538 {
```

```
539 if (vflag == 1 || vflag ==888)
```

```
540 {  
541     memory_leak_001();  
542 }
```

```
}
```

01.w_Defects/memory_leak.c:219

```
216 *(p+1) = 1;  
217 free(ptr);  
218 }
```

```
219 }
```

```
220  
221 /*
```

```
222 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

01.w_Defects/memory_leak.c:236

Level Medium

Status Not processed

```
233 free (ptr);  
234 ptr = NULL;  
235 }
```

```
236 }
```

```
237  
238 /*
```

```
239 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

Trace

```
vflag == 1
```

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;
537 void memory_leak_main ()
538 {
539   if (vflag == 1 || vflag == 888)
540   {
541     memory_leak_001();
542 }
```

```
}
```

01.w_Defects/memory_leak.c:236

```
233   free (ptr);
234   ptr = NULL;
235 }

236 }

237
238 /*
239 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

01.w_Defects/memory_leak.c:248

Level Medium

Status Not processed

```
245 int *p1 = (int*) calloc(5,sizeof(int)); /*Tool should detect this line as error*/ /*ERROR:
Memory Leakage */
246 int *p2 = NULL;
247 p1 = ptr;

248 p2 = p1;
```

```
249 *(p2+4) = 1;  
250 free(ptr);  
251 }
```

Trace

vflag == 1

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;  
537 void memory_leak_main ()  
538 {  
  
539 if (vflag == 1 || vflag ==888)  
  
540 {  
541     memory_leak_001();  
542 }
```

p2 = p1

01.w_Defects/memory_leak.c:248

```
245 int *p1 = (int*) calloc(5,sizeof(int));/*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
246 int *p2 = NULL;  
247 p1 = ptr;  
  
248 p2 = p1;  
  
249 *(p2+4) = 1;  
250 free(ptr);  
251 }
```

01.w_Defects/memory_leak.c:276

Level Medium

Status Not processed

```
273  {  
274      char * buf ;  
275  
276      buf = un.u2;  
277  }  
278 }  
279
```

Trace

```
vflag == 1
```

```
01.w_Defects/memory_leak.c:539
```

```
536 extern volatile int vflag;  
537 void memory_leak_main ()  
538 {  
  
539     if (vflag == 1 || vflag ==888)  
  
540     {  
541         memory_leak_001();  
542     }
```

```
buf = un.u2
```

```
01.w_Defects/memory_leak.c:276
```

```
273  {  
274      char * buf ;  
275  
276      buf = un.u2;  
277  }  
278 }  
279
```

01.w_Defects/memory_leak.c:311

Level Medium**Status** Not processed

```
308 memory_leak_0012_uni_001 *p = (memory_leak_0012_uni_001 *)malloc(5*sizeof( memory_leak_0012_uni_001 ));/*Tool should detect this line as error*/ /*ERROR:  
Memory Leakage */  
309 p = u;  
310  
  
311 p->s1.a = 1;  
  
312  
313 free(u);  
314 }
```

Trace

vflag == 1

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;  
537 void memory_leak_main ()  
538 {  
  
539 if (vflag == 1 || vflag ==888)  
  
540 {  
541     memory_leak_001();  
542 }
```

p->s1.a = 1

01.w_Defects/memory_leak.c:311

```
308 memory_leak_0012_uni_001 *p = (memory_leak_0012_uni_001 *)malloc(5*sizeof( memory_leak_0012_uni_001 ));/*Tool should detect this line as error*/ /*ERROR:Memory Leakage */  
309 p = u;  
310
```

```
311 p->s1.a = 1;
```

```
312  
313 free(u);  
314 }
```

01.w_Defects/memory_leak.c:359

Level Medium

Status Not processed

```
356 free(p);  
357  
358 }
```

```
359 }
```

```
360  
361 /*  
362 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

Trace

```
vflag == 1
```

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;  
537 void memory_leak_main ()  
538 {
```

```
539 if (vflag == 1 || vflag == 888)
```

```
540 {  
541     memory_leak_001();  
542 }
```

```
}
```

01.w_Defects/memory_leak.c:359

```
356 free(p);  
357  
358 }
```

```
359 }
```

```
360  
361 /*  
362 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

01.w_Defects/memory_leak.c:384

Level Medium

Status Not processed

```
381     float * fptr1 ;  
382     fptr1 = *fp2;  
383 }
```

```
384 }
```

```
385  
386 /*  
387 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

Trace

```
vflag == 1
```

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;  
537 void memory_leak_main ()  
538 {
```

```
539 if (vflag == 1 || vflag ==888)
```

```
540 {  
541     memory_leak_001();  
542 }
```

```
}
```

01.w_Defects/memory_leak.c:384

```
381     float * fptr1 ;  
382     fptr1 = *fp2;  
383 }
```

```
384 }
```

```
385  
386 /*
```

```
387 * Types of defects: Memory Leakage - Allocate Memory and not freeing it
```

01.w_Defects/memory_leak.c:404

Level Medium

Status Not processed

```
401 {  
402     strcpy(str1,str);  
403 }
```

```
404 }
```

```
405
```

```
406 /*
```

```
407 * Types of defects: Freeing a NULL pointer
```

Trace

```
vflag == 1
```

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;
537 void memory_leak_main ()
538 {
539     if (vflag == 1 || vflag == 888)
540     {
541         memory_leak_001();
542     }
}
```

01.w_Defects/memory_leak.c:404

```
401 {
402     strcpy(str1,str);
403 }

404 }

405
406 /*
407 * Types of defects: Freeing a NULL pointer

```

01.w_Defects/uninit_memory_access.c:201

Level Medium

Status Not processed

```
198 uninit_memory_access_008_s_001 *s = NULL;
199 s = uninit_memory_access_008_func_001();
200 s->b = s->a; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Memory
Access*/
201 }
```

```
202
203 /*
204 * Types of defects: Uninitialized Memory Access
```

Trace

```
vflag == 1
```

```
01.w_Defects/uninit_memory_access.c:462
```

```
459 extern volatile int vflag;
460 void uninit_memory_access_main ()
461 {
462     if (vflag == 1 || vflag == 888)
463     {
464         uninit_memory_access_001();
465     }
}
```

```
01.w_Defects/uninit_memory_access.c:201
```

```
198     uninit_memory_access_008_s_001 *s = NULL;
199     s = uninit_memory_access_008_func_001();
200     s->b = s->a; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
Memory Access*/
```

```
201 }
```

```
202
203 /*
204 * Types of defects: Uninitialized Memory Access
```

```
01.w_Defects/uninit_pointer.c:125
```

Level Medium

Status Not processed

```
122 char *buf3=strdup("String3");
123 char *buf4=strdup("String4");
124 char *buf5=strdup("String5");

125     if (!buf1 || !buf3 || !buf4 || !buf5) return;

126 char **pbuf[5] = {&buf2, &buf3, &buf4, &buf5};
127 int i,j=4;
128
```

Trace

```
vflag == 1
```

```
01.w_Defects/uninit_pointer.c:421
```

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {

421     if (vflag == 1 || vflag ==888)

422 {
423     uninit_pointer_001();
424 }
```

```
if (!buf1 || !buf3 || !buf4 || !buf5) return
```

```
01.w_Defects/uninit_pointer.c:125
```

```
122 char *buf3=strdup("String3");
123 char *buf4=strdup("String4");
124 char *buf5=strdup("String5");

125     if (!buf1 || !buf3 || !buf4 || !buf5) return;

126 char **pbuf[5] = {&buf2, &buf3, &buf4, &buf5};
127 int i,j=4;
128
```

01.w_Defects/uninit_pointer.c:125

Level Medium**Status** Not processed

```
122 char *buf3=strdup("String3");
123 char *buf4=strdup("String4");
124 char *buf5=strdup("String5");

125     if (!buf1 || !buf3 || !buf4 || !buf5) return;

126 char **pbuff[5] = {&buf2, &buf3, &buf4, &buf5};
127 int i,j=4;
128
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {

421     if (vflag == 1 || vflag ==888)

422     {
423         uninit_pointer_001();
424     }
```

if (!buf1 || !buf3 || !buf4 || !buf5) return

01.w_Defects/uninit_pointer.c:125

```
122 char *buf3=strdup("String3");
123 char *buf4=strdup("String4");
124 char *buf5=strdup("String5");

125     if (!buf1 || !buf3 || !buf4 || !buf5) return;

126 char **pbuff[5] = {&buf2, &buf3, &buf4, &buf5};
```

```
127 int i,j=4;  
128
```

01.w_Defects/uninit_pointer.c:125

Level Medium

Status Not processed

```
122 char *buf3=strdup("String3");  
123 char *buf4=strdup("String4");  
124 char *buf5=strdup("String5");
```

```
125 if (!buf1 || !buf3 || !buf4 || !buf5) return;
```

```
126 char **pbuf[5] = {&buf2, &buf3, &buf4, &buf5};  
127 int i,j=4;  
128
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;  
419 void uninit_pointer_main ()  
420 {
```

```
421 if (vflag == 1 || vflag ==888)
```

```
422 {  
423     uninit_pointer_001();  
424 }
```

```
if (!buf1 || !buf3 || !buf4 || !buf5) return
```

01.w_Defects/uninit_pointer.c:125

```
122 char *buf3=strdup("String3");
123 char *buf4=strdup("String4");
124 char *buf5=strdup("String5");
```

```
125     if (!buf1 || !buf3 || !buf4 || !buf5) return;
```

```
126 char **pbuf[5] = {&buf2, &buf3, &buf4, &buf5};
127 int i,j=4;
128
```

01.w_Defects/uninit_pointer.c:125

Level Medium

Status Not processed

```
122 char *buf3=strdup("String3");
123 char *buf4=strdup("String4");
124 char *buf5=strdup("String5");
```

```
125     if (!buf1 || !buf3 || !buf4 || !buf5) return;
```

```
126 char **pbuf[5] = {&buf2, &buf3, &buf4, &buf5};
127 int i,j=4;
128
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {
```

```
421     if (vflag == 1 || vflag ==888)
```

```
422 {  
423     uninit_pointer_001();  
424 }
```

```
if (!buf1 || !buf3 || !buf4 || !buf5) return
```

01.w_Defects/uninit_pointer.c:125

```
122 char *buf3=strdup("String3");  
123 char *buf4=strdup("String4");  
124 char *buf5=strdup("String5");
```

```
125     if (!buf1 || !buf3 || !buf4 || !buf5) return;
```

```
126 char **pbuf[5] = {&buf2, &buf3, &buf4, &buf5};  
127 int i,j=4;  
128
```

01.w_Defects/uninit_pointer.c:236

Level Medium

Status Not processed

```
233     break;  
234 }  
235 }
```

```
236 }
```

```
237  
238 /*  
239 * Types of defects: Uninitialized pointer
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {
421     if (vflag == 1 || vflag == 888)
422     {
423         uninit_pointer_001();
424     }
}
```

01.w_Defects/uninit_pointer.c:236

```
233     break;
234 }
235 }

236 }

237
238 /*
239 * Types of defects: Uninitialized pointer
```

01.w_Defects/wrong_arguments_func_pointer.c:601

Level Medium

Status Not processed

```
598 void (*fptr3)(wrong_arguments_func_pointer_018_s_001 st,
wrong_arguments_func_pointer_018_s_001* st1);
599 fptr3 = wrong_arguments_func_pointer_018_func_004;
600 fptr3(st,st1);

601 }
```

```
602
603 /*
604 * Type of defect: Wrong arguments passed to a function pointer
```

Trace

```
vflag == 1
```

```
01.w_Defects/wrong_arguments_func_pointer.c:610
```

```
607 extern volatile int vflag;
608 void wrong_arguments_func_pointer_main ()
609 {
610     if (vflag == 1 || vflag == 888)
611     {
612         wrong_arguments_func_pointer_001();
613     }
}
```

```
01.w_Defects/wrong_arguments_func_pointer.c:601
```

```
598 void (*fptr3)(wrong_arguments_func_pointer_018_s_001 st,
599                 wrong_arguments_func_pointer_018_s_001* st1);
600 fptr3 = wrong_arguments_func_pointer_018_func_004;
601 fptr3(st, st1);
```

```
601 }
```

```
602
603 /*
604 * Type of defect: Wrong arguments passed to a function pointer
```

```
02.wo_Defects/free_null_pointer.c:491
```

Level Medium

Status Not processed

```
488 }
489 *(fptr+3) = 50.5;
490 *fp1 = fptr;

491 i++;

492 }while(i>=0 && i<=1);
493 do
494 {
```

Trace

```
vflag == 1
```

```
02.wo_Defects/free_null_pointer.c:572
```

```
569 extern volatile int vflag;
570 void free_null_pointer_main ()
571 {

572 if (vflag == 1 || vflag ==888)

573 {
574     free_null_pointer_001();
575 }
```

```
i++
```

```
02.wo_Defects/free_null_pointer.c:491
```

```
488 }
489 *(fptr+3) = 50.5;
490 *fp1 = fptr;

491 i++;

492 }while(i>=0 && i<=1);
493 do
494 {
```

02.wo_Defects/func_pointer.c:635

Level Medium

Status Not processed

```
632 void (*fptr3)(func_pointer_015_s_001* st1);
633 fptr3 = func_pointer_015_func_004;
634 fptr3(st1);
```

635 }

```
636
637 /*
638 * Type of defect: bad function pointer casting
```

Trace

vflag == 1

02.wo_Defects/func_pointer.c:644

```
641 extern volatile int vflag;
642 void func_pointer_main ()
643 {
```

644 if (vflag == 1 || vflag == 888)

```
645 {
646     func_pointer_001();
647 }
```

}

02.wo_Defects/func_pointer.c:635

```
632 void (*fptr3)(func_pointer_015_s_001* st1);
633 fptr3 = func_pointer_015_func_004;
634 fptr3(st1);
```

635 }

636

```
637 /*  
638 * Type of defect: bad function pointer casting
```

02.wo_Defects/memory_allocation_failure.c:293

Level Medium

Status Not processed

```
290 do  
291 {  
292     buf = (char*) malloc(MAX_BUFFER * sizeof(char)); /*Tool should not detect  
this line as error*/ /*No ERROR:Memory allocation failure */  
  
293     i++;  
  
294 }while (i<MAX_VAL);  
295 }  
296
```

Trace

```
vflag == 1
```

02.wo_Defects/memory_allocation_failure.c:741

```
738 extern volatile int vflag;  
739 void memory_allocation_failure_main ()  
740 {  
  
741     if (vflag == 1 || vflag ==888)  
  
742     {  
743         memory_allocation_failure_001 ();  
744     }
```

i++

02.wo_Defects/memory_allocation_failure.c:293

```
290     do
291     {
292         buf = (char*) malloc(MAX_BUFFER * sizeof(char)); /*Tool should not
detect this line as error*/ /*No ERROR:Memory allocation failure */

293         i++;

294     }while (i<MAX_VAL);
295 }
296
```

02.wo_Defects/uninit_memory_access.c:131

Level Medium**Status** Not processed

```
128     char *str2 = "STRING";
129     uninit_memory_access_006_func_001(str1, str2);
130     printf("%s\n", str1);/*Tool should not detect this line as error*/ /*No ERROR:
Uninitialized Memory Access*/

131 }

132
133 /*
134 * Types of defects: Uninitialized Memory Access
```

Trace

vflag == 1

02.wo_Defects/uninit_memory_access.c:483

```
480 extern volatile int vflag;
481 void uninit_memory_access_main ()
482 {
```

```
483 if (vflag == 1 || vflag ==888)

484 {
485     uninit_memory_access_001();
486 }
```

```
}
```

02.wo_Defects/uninit_memory_access.c:131

```
128 char *str2 = "STRING";
129 uninit_memory_access_006_func_001(str1, str2);
130 printf("%s\n", str1);/*Tool should not detect this line as error*/ /*No ERROR:
Uninitialized Memory Access*/
```

```
131 }
```

```
132
133 /*
134 * Types of defects: Uninitialized Memory Access
```

02.wo_Defects/uninit_pointer.c:132

Level Medium

Status Not processed

```
129 char *buf3=strdup("String3");
130 char *buf4=strdup("String4");
131 char *buf5=strdup("String5");

132     if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;

133 char **pbuf[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};
134 int i,j=4;
135
```

Trace

```
vflag == 1
```

02.wo_Defects/uninit_pointer.c:440

```
437 extern volatile int vflag;
438 void uninit_pointer_main ()
439 {
440     if (vflag == 1 || vflag ==888)
441     {
442         uninit_pointer_001 ();
443     }
```

```
if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return
```

02.wo_Defects/uninit_pointer.c:132

```
129 char *buf3=strdup("String3");
130 char *buf4=strdup("String4");
131 char *buf5=strdup("String5");

132     if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;

133 char **pbuff[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};
134 int i,j=4;
135
```

02.wo_Defects/uninit_pointer.c:132

Level Medium

Status Not processed

```
129 char *buf3=strdup("String3");
130 char *buf4=strdup("String4");
131 char *buf5=strdup("String5");
```

```
132     if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;
```

```
133 char **pbuff[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};
```

```
134 int i,j=4;  
135
```

Trace

```
vflag == 1
```

02.wo_Defects/uninit_pointer.c:440

```
437 extern volatile int vflag;  
438 void uninit_pointer_main ()  
439 {  
  
440 if (vflag == 1 || vflag ==888)  
  
441 {  
442     uninit_pointer_001 ();  
443 }
```

```
if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return
```

02.wo_Defects/uninit_pointer.c:132

```
129 char *buf3=strdup("String3");  
130 char *buf4=strdup("String4");  
131 char *buf5=strdup("String5");  
  
132 if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;  
  
133 char **pbuff[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};  
134 int i,j=4;  
135
```

02.wo_Defects/uninit_pointer.c:132

Level Medium

Status Not processed

```
129 char *buf3=strdup("String3");
```

```
130 char *buf4=strdup("String4");
131 char *buf5=strdup("String5");

132     if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;

133 char **pbuf[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};
134 int i,j=4;
135
```

Trace

```
vflag == 1
```

```
02.wo_Defects/uninit_pointer.c:440
```

```
437 extern volatile int vflag;
438 void uninit_pointer_main ()
439 {

440     if (vflag == 1 || vflag ==888)

441 {
442     uninit_pointer_001 ();
443 }
```

```
if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return
```

```
02.wo_Defects/uninit_pointer.c:132
```

```
129 char *buf3=strdup("String3");
130 char *buf4=strdup("String4");
131 char *buf5=strdup("String5");

132     if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;

133 char **pbuf[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};
134 int i,j=4;
135
```

02.wo_Defects/uninit_pointer.c:132

Level Medium**Status** Not processed

```
129 char *buf3=strdup("String3");
130 char *buf4=strdup("String4");
131 char *buf5=strdup("String5");

132     if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;

133 char **pbuf[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};
134 int i,j=4;
135
```

Trace

vflag == 1

02.wo_Defects/uninit_pointer.c:440

```
437 extern volatile int vflag;
438 void uninit_pointer_main ()
439 {

440     if (vflag == 1 || vflag ==888)

441     {
442         uninit_pointer_001 ();
443     }
```

if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return

02.wo_Defects/uninit_pointer.c:132

```
129 char *buf3=strdup("String3");
130 char *buf4=strdup("String4");
131 char *buf5=strdup("String5");

132     if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;
```

```
133 char **pbuf[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};
```

```
134 int i,j=4;  
135
```

02.wo_Defects/uninit_pointer.c:132

Level Medium

Status Not processed

```
129 char *buf3=strdup("String3");  
130 char *buf4=strdup("String4");  
131 char *buf5=strdup("String5");
```

```
132 if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;
```

```
133 char **pbuf[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};  
134 int i,j=4;  
135
```

Trace

```
vflag == 1
```

02.wo_Defects/uninit_pointer.c:440

```
437 extern volatile int vflag;  
438 void uninit_pointer_main ()  
439 {
```

```
440 if (vflag == 1 || vflag ==888)
```

```
441 {  
442     uninit_pointer_001 ();  
443 }
```

```
if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return
```

02.wo_Defects/uninit_pointer.c:132

```
129 char *buf3=strdup("String3");
130 char *buf4=strdup("String4");
131 char *buf5=strdup("String5");
```

```
132     if (!buf1 || !buf2 || !buf3 || !buf4 || !buf5) return;
```

```
133 char **pbuff[5] = {&buf1, &buf2, &buf3, &buf4, &buf5};
134 int i,j=4;
135
```

02.wo_Defects/uninit_pointer.c:248

Level Medium

Status Not processed

```
245     break;
246 }
247 }
```

```
248 }
```

```
249
250 /*
251 * Types of defects: Uninitialized pointer
```

Trace

```
vflag == 1
```

02.wo_Defects/uninit_pointer.c:440

```
437 extern volatile int vflag;
438 void uninit_pointer_main ()
439 {
```

```
440     if (vflag == 1 || vflag ==888)
```

```
441 {  
442     uninit_pointer_001 ();  
443 }
```

```
}
```

02.wo_Defects/uninit_pointer.c:248

```
245     break;  
246 }  
247 }
```

```
248 }
```

```
249  
250 /*  
251 * Types of defects: Uninitialized pointer
```

02.wo_Defects/wrong_arguments_func_pointer.c:218

Level Medium

Status Not processed

```
215 char *str1 = strdup("STRING33");  
216 if (!str1) return;  
217 char *str2 = strdup("STRING55");
```

```
218 if (!str2) return;
```

```
219 char *str3 = (char *) malloc(20*sizeof(char));  
220 if (!str3) return;  
221 char ret;
```

Trace

```
vflag == 1
```

02.wo_Defects/wrong_arguments_func_pointer.c:605

```
602 extern volatile int vflag;
603 void wrong_arguments_func_pointer_main ()
604 {
605     if (vflag == 1 || vflag == 888)
606     {
607         wrong_arguments_func_pointer_001 ();
608     }
}
```

```
if (!str2) return
```

02.wo_Defects/wrong_arguments_func_pointer.c:218

```
215 char *str1 = strdup("STRING33");
216 if (!str1) return;
217 char *str2 = strdup("STRING55");

218 if (!str2) return;

219 char *str3 = (char *) malloc(20 * sizeof(char));
220 if (!str3) return;
221 char ret;
```

02.wo_Defects/wrong_arguments_func_pointer.c:220

Level Medium

Status Not processed

```
217 char *str2 = strdup("STRING55");
218 if (!str2) return;
219 char *str3 = (char *) malloc(20 * sizeof(char));

220 if (!str3) return;

221 char ret;
```

```
222 char (*func)(char *,char *, char *);
223 func = wrong_arguments_func_pointer_009_func_001;
```

Trace

```
vflag == 1
```

```
02.wo_Defects/wrong_arguments_func_pointer.c:605
```

```
602 extern volatile int vflag;
603 void wrong_arguments_func_pointer_main ()
604 {

605 if (vflag == 1 || vflag ==888)

606 {
607     wrong_arguments_func_pointer_001 ();
608 }
```

```
if (!str3) return
```

```
02.wo_Defects/wrong_arguments_func_pointer.c:220
```

```
217 char *str2 = strdup("STRING55");
218 if (!str2) return;
219 char *str3 = (char *) malloc(20*sizeof(char));

220 if (!str3) return;

221 char ret;
222 char (*func)(char *,char *, char *);
223 func = wrong_arguments_func_pointer_009_func_001;
```

```
02.wo_Defects/wrong_arguments_func_pointer.c:357
```

Level Medium

Status Not processed

```
354 fptr1 = wrong_arguments_func_pointer_012_func_002;
```

```
355 fptr1(st,st1);
356

357 void (*fptr2)(wrong_arguments_func_pointer_012_s_001 *,int);

358 fptr2 = wrong_arguments_func_pointer_012_func_003;
359 fptr2(&st,1);
360 }
```

Trace

vflag == 1

02.wo_Defects/wrong_arguments_func_pointer.c:605

```
602 extern volatile int vflag;
603 void wrong_arguments_func_pointer_main ()
604 {

605 if (vflag == 1 || vflag ==888)

606 {
607     wrong_arguments_func_pointer_001 ();
608 }
```

void (*fptr2)
(wrong_arguments_func_pointer_012_s_00
1 *,int)

02.wo_Defects/wrong_arguments_func_pointer.c:357

```
354 fptr1 = wrong_arguments_func_pointer_012_func_002;
355 fptr1(st,st1);
356

357 void (*fptr2)(wrong_arguments_func_pointer_012_s_001 *,int);

358 fptr2 = wrong_arguments_func_pointer_012_func_003;
359 fptr2(&st,1);
360 }
```

02.wo_Defects/wrong_arguments_func_pointer.c:596

Level Medium**Status** Not processed

```
593 void (*fptr3)(wrong_arguments_func_pointer_018_s_001 st,  
wrong_arguments_func_pointer_018_s_001* st1);  
594 fptr3 = wrong_arguments_func_pointer_018_func_004;  
595 fptr3(st,st1);
```

596 }

597

598 /*

599 * Type of defect: Wrong arguments passed to a function pointer

Trace

vflag == 1

02.wo_Defects/wrong_arguments_func_pointer.c:605

```
602 extern volatile int vflag;  
603 void wrong_arguments_func_pointer_main ()  
604 {
```

605 if (vflag == 1 || vflag ==888)

```
606 {  
607     wrong_arguments_func_pointer_001 ();  
608 }
```

}

02.wo_Defects/wrong_arguments_func_pointer.c:596

```
593 void (*fptr3)(wrong_arguments_func_pointer_018_s_001 st,  
wrong_arguments_func_pointer_018_s_001* st1);  
594 fptr3 = wrong_arguments_func_pointer_018_func_004;  
595 fptr3(st,st1);
```

596 }

```
597
598 /*
599 * Type of defect: Wrong arguments passed to a function pointer
```

Mutex double lock (C/C++)

Description

In programming, mutexes (mutual exclusions) are used to synchronize simultaneously running threads, and only the thread that owns the mutex can release it, i.e. transfer to the marked state.

The purpose of the mutex is to protect the object from access to other threads, other than the one that owns the mutex. In this case, only one thread can own an object protected by a mutex at any particular time.

Trying to lock mutex that is already locked. This will cause undefined behavior.

Example

In the following example the program tries to lock the same mutex twice:

```
pthread_mutex_t mtx1;

void
f()
{
    pthread_mutex_lock(&mtx1);
    pthread_mutex_lock(&mtx1); // pthread double lock
}
```

Recommendations

- Be careful not to lock a mutex that has already been locked.

Links

1. Mutual exclusion
2. pthread_mutex_lock

Vulnerability Entries

01.w_Defects/double_lock.c:42

Level Medium

Status Not processed

```
39 int ip = (int)pthread_self();
40 pthread_mutex_lock(&double_lock_001_glb_mutex);
41 double_lock_001_glb_data = (double_lock_001_glb_data % 100) + 1;

42 pthread_mutex_lock(&double_lock_001_glb_mutex); /*Tool should detect this line
as error*/ /*ERROR:Double Lock*/

43 double_lock_001_glb_data = (double_lock_001_glb_data % 100) + 1;
44
45 printf("Task1! It's me, thread #%d!\n",ip);
```

Trace

&double_lock_001_glb_mutex

01.w_Defects/double_lock.c:42

```
39 int ip = (int)pthread_self();
40 pthread_mutex_lock(&double_lock_001_glb_mutex);
41 double_lock_001_glb_data = (double_lock_001_glb_data % 100) + 1;
```

```
42 pthread_mutex_lock(&double_lock_001_glb_mutex); /*Tool should detect
this line as error*/ /*ERROR:Double Lock*/
```

```
43 double_lock_001_glb_data = (double_lock_001_glb_data % 100) + 1;
44
45 printf("Task1! It's me, thread #%d!\n",ip);
```

&double_lock_001_glb_mutex

01.w_Defects/double_lock.c:42

```
39 int ip = (int)pthread_self();
40 pthread_mutex_lock(&double_lock_001_glb_mutex);
41 double_lock_001_glb_data = (double_lock_001_glb_data % 100) + 1;

42 pthread_mutex_lock(&double_lock_001_glb_mutex); /*Tool should detect
this line as error*/ /*ERROR:Double Lock*/

43 double_lock_001_glb_data = (double_lock_001_glb_data % 100) + 1;
44
45 printf("Task1! It's me, thread #%d!\n",ip);
```

01.w_Defects/double_lock.c:93

Level Medium

Status Not processed

```
90 #if ! defined(CHECKER_POLYSPACE)
91   pthread_mutex_lock (&double_lock_002_glb_mutex);
92   double_lock_002_glb_data = (double_lock_002_glb_data% 100) + 1;

93   pthread_mutex_lock (&double_lock_002_glb_mutex); /*Tool should detect this line
as error*/ /*ERROR:Double Lock*/

94   double_lock_002_glb_data = (double_lock_002_glb_data% 100) + 1;
95   pthread_mutex_unlock(&double_lock_002_glb_mutex);
96 #endif /* defined(CHECKER_POLYSPACE) */
```

Trace

&double_lock_002_glb_mutex

01.w_Defects/double_lock.c:93

```
90 #if ! defined(CHECKER_POLYSPACE)
91   pthread_mutex_lock (&double_lock_002_glb_mutex);
92   double_lock_002_glb_data = (double_lock_002_glb_data% 100) + 1;
```

```
93 pthread_mutex_lock (&double_lock_002_glb_mutex); /*Tool should detect  
this line as error*/ /*ERROR:Double Lock*/  
  
94 double_lock_002_glb_data = (double_lock_002_glb_data% 100) + 1;  
95 pthread_mutex_unlock(&double_lock_002_glb_mutex);  
96 #endif /* defined(CHECKER_POLYSPACE) */
```

&double_lock_002_glb_mutex

01.w_Defects/double_lock.c:93

```
90 #if ! defined(CHECKER_POLYSPACE)  
91 pthread_mutex_lock (&double_lock_002_glb_mutex);  
92 double_lock_002_glb_data = (double_lock_002_glb_data% 100) + 1;  
  
93 pthread_mutex_lock (&double_lock_002_glb_mutex); /*Tool should detect  
this line as error*/ /*ERROR:Double Lock*/  
  
94 double_lock_002_glb_data = (double_lock_002_glb_data% 100) + 1;  
95 pthread_mutex_unlock(&double_lock_002_glb_mutex);  
96 #endif /* defined(CHECKER_POLYSPACE) */
```

01.w_Defects/double_lock.c:140

Level Medium

Status Not processed

```
137 void double_lock_003_func_001 ()  
138 {  
139 #if ! defined(CHECKER_POLYSPACE)
```

```
140 pthread_mutex_lock (&double_lock_003_glb_mutex); /*Tool should detect this line  
as error*/ /*ERROR:Double Lock*/
```

```
141 double_lock_003_glb_data = (double_lock_003_glb_data% 100) + 1;  
142 /*pthread_mutex_unlock (&double_lock_003_glb_mutex);*/  
143 #endif /* ! defined(CHECKER_POLYSPACE) */
```

Trace

double_lock_003_func_001 ()

01.w_Defects/double_lock.c:152

```
149 pthread_mutex_lock (&double_lock_003_glb_mutex);
150 double_lock_003_glb_data = (double_lock_003_glb_data% 100) + 1;
151

152 double_lock_003_func_001 ();

153
154 /*pthread_mutex_unlock (&double_lock_003_glb_mutex);*/
155 #endif /* ! defined(CHECKER_POLYSPACE) */
```

&double_lock_003_glb_mutex

01.w_Defects/double_lock.c:140

```
137 void double_lock_003_func_001 ()
138 {
139 #if ! defined(CHECKER_POLYSPACE)

140 pthread_mutex_lock (&double_lock_003_glb_mutex); /*Tool should detect
this line as error*/ /*ERROR:Double Lock*/

141 double_lock_003_glb_data = (double_lock_003_glb_data% 100) + 1;
142 /*pthread_mutex_unlock (&double_lock_003_glb_mutex);*/
143 #endif /* ! defined(CHECKER_POLYSPACE) */
```

01.w_Defects/livelock.c:32

Level Medium**Status** Not processed

```
29 x=x+1;
30 pthread_mutex_unlock(&livelock_001_glb_A);
31
```

```
32 int status(pthread_mutex_trylock(&livelock_001_glb_B); /*Tool should detect this line
as error*/ /*ERROR: Live lock*/
```

```
33 if(status==0)
34 {
35   continue;
```

Trace

&livelock_001_glb_B

01.w_Defects/livelock.c:32

```
29 x=x+1;
30 pthread_mutex_unlock(&livelock_001_glb_A);
31
```

32 int status(pthread_mutex_trylock(&livelock_001_glb_B); /*Tool should detect
this line as error*/ /*ERROR: Live lock*/

```
33 if(status==0)
34 {
35   continue;
```

&livelock_001_glb_B

01.w_Defects/livelock.c:32

```
29 x=x+1;
30 pthread_mutex_unlock(&livelock_001_glb_A);
31
```

32 int status(pthread_mutex_trylock(&livelock_001_glb_B); /*Tool should detect
this line as error*/ /*ERROR: Live lock*/

```
33 if(status==0)
34 {
35   continue;
```

01.w_Defects/livelock.c:49

Level Medium

Status Not processed

```
46 y=y+1;
47 pthread_mutex_unlock(&livelock_001_glb_B);
48

49 int status(pthread_mutex_trylock(&livelock_001_glb_A));

50 if(status==0)
51 {
52     continue;
```

Trace

```
&livelock_001_glb_A
```

```
01.w_Defects/livelock.c:49
```

```
46 y=y+1;
47 pthread_mutex_unlock(&livelock_001_glb_B);
48

49 int status(pthread_mutex_trylock(&livelock_001_glb_A));

50 if(status==0)
51 {
52     continue;
```

```
&livelock_001_glb_A
```

```
01.w_Defects/livelock.c:49
```

```
46 y=y+1;
47 pthread_mutex_unlock(&livelock_001_glb_B);
48

49 int status(pthread_mutex_trylock(&livelock_001_glb_A));

50 if(status==0)
51 {
52     continue;
```

01.w_Defects/lock_never_unlock.c:93

Level Medium**Status** Not processed

```
90 pthread_mutex_lock(&lock_never_unlock_002_glb_mutex);
91 lock_never_unlock_002_glb_data = (lock_never_unlock_002_glb_data % 100) + 1;
92 /*Tool should detect this line as error*/ /* ERROR:Lock Never Unlock */

93 pthread_mutex_lock(&lock_never_unlock_002_glb_mutex);

94 lock_never_unlock_002_glb_data = (lock_never_unlock_002_glb_data % 100) + 1;
95 #if defined PRINT_DEBUG
96 unsigned long ip = (unsigned long)pthread_self();
```

Trace

&lock_never_unlock_002_glb_mutex

01.w_Defects/lock_never_unlock.c:93

```
90 pthread_mutex_lock(&lock_never_unlock_002_glb_mutex);
91 lock_never_unlock_002_glb_data = (lock_never_unlock_002_glb_data %
100) + 1;
92 /*Tool should detect this line as error*/ /* ERROR:Lock Never Unlock */

93 pthread_mutex_lock(&lock_never_unlock_002_glb_mutex);

94 lock_never_unlock_002_glb_data = (lock_never_unlock_002_glb_data %
100) + 1;
95 #if defined PRINT_DEBUG
96 unsigned long ip = (unsigned long)pthread_self();
```

&lock_never_unlock_002_glb_mutex

01.w_Defects/lock_never_unlock.c:93

```
90 pthread_mutex_lock(&lock_never_unlock_002_glb_mutex);
91 lock_never_unlock_002_glb_data = (lock_never_unlock_002_glb_data %
100) + 1;
92 /*Tool should detect this line as error*/ /* ERROR:Lock Never Unlock */

93 pthread_mutex_lock(&lock_never_unlock_002_glb_mutex);
```

```
94     lock_never_unlock_002_glb_data = (lock_never_unlock_002_glb_data %  
100) + 1;  
95 #if defined PRINT_DEBUG  
96     unsigned long ip = (unsigned long)pthread_self();
```

01.w_Defects/lock_never_unlock.c:147

Level Medium

Status Not processed

```
144 void lock_never_unlock_003_func_001 (void *pram)  
145 {  
146 #if ! defined(CHECKER_POLYSPACE)  
  
147     pthread_mutex_lock (&lock_never_unlock_003_glb_mutex);  
  
148     lock_never_unlock_003_glb_data = (lock_never_unlock_003_glb_data) + 1.2;  
149     /*Tool should detect this line as error*/ /* ERROR:Lock Never Unlock */  
150 #if defined PRINT_DEBUG
```

Trace

lock_never_unlock_003_func_001(pram)

01.w_Defects/lock_never_unlock.c:163

```
160  
161     pthread_mutex_lock(&lock_never_unlock_003_glb_mutex);  
162     lock_never_unlock_003_glb_data = (lock_never_unlock_003_glb_data) +  
3.5;  
  
163     lock_never_unlock_003_func_001(pram);  
  
164     pthread_mutex_unlock(&lock_never_unlock_003_glb_mutex);  
165 #endif /* defined(CHECKER_POLYSPACE) */  
166     return NULL;
```

&lock_never_unlock_003_glb_mutex

01.w_Defects/lock_never_unlock.c:147

```
144 void lock_never_unlock_003_func_001 (void *pram)
145 {
146 #if ! defined(CHECKER_POLYSPACE)

147     pthread_mutex_lock (&lock_never_unlock_003_glb_mutex);

148     lock_never_unlock_003_glb_data = (lock_never_unlock_003_glb_data) +
149     1.2;
150 /*Tool should detect this line as error*/ /* ERROR:Lock Never Unlock */
150 #if defined PRINT_DEBUG
```

01.w_Defects/lock_never_unlock.c:398

Level Medium**Status** Not processed

```
395 {
396     if (ip >= 0)
397     {

398         pthread_mutex_lock( &lock_never_unlock_007_glb_mutex_2 );

399
400         if(i!=5)
401         {
```

Trace

ip >= 0

01.w_Defects/lock_never_unlock.c:396

```
393     ip = ip *20;
394     while (i>0)
395 {
```

```
396     if (ip >= 0)

397     {
398         pthread_mutex_lock( &lock_never_unlock_007_glb_mutex_2 );
399     }
```

&lock_never_unlock_007_glb_mutex_2

01.w_Defects/lock_never_unlock.c:398

```
395 {
396     if (ip >= 0)
397     {

398         pthread_mutex_lock( &lock_never_unlock_007_glb_mutex_2 );

399
400         if(i!=5)
401     {
```

01.w_Defects/lock_never_unlock.c:551

Level Medium

Status Not processed

```
548 {
549     if (ip >= 0)
550     {

551         pthread_mutex_lock( &lock_never_unlock_009_glb_mutex_2 );

552
553         if(i!=5)
554     {
```

Trace

```
ip >= 0
```

01.w_Defects/lock_never_unlock.c:549

```
546 ip = ip *20;
547 do
548 {

549 if (ip >= 0)

550 {
551     pthread_mutex_lock( &lock_never_unlock_009_glb_mutex_2 );
552 }
```

```
&lock_never_unlock_009_glb_mutex_2
```

01.w_Defects/lock_never_unlock.c:551

```
548 {
549 if (ip >= 0)
550 {

551     pthread_mutex_lock( &lock_never_unlock_009_glb_mutex_2 );

552
553     if(i!=5)
554 {
```

Mutex double unlock (C/C++)

Description

In programming, mutexes (mutual exclusions) are used to synchronize simultaneously running threads, and only the thread that owns the mutex can release it, i.e. transfer to the marked state.

The purpose of the mutex is to protect the object from access to other threads, other than the one that owns the mutex. In this case, only one thread can own an object protected by a mutex at any particular time.

Trying to unlock mutex that is already unlocked. This will cause undefined behavoir.

Example

In the following example the program tries to unlock the same mutex twice:

```
pthread_mutex_t mtx1;
```

```
void
f()
{
    pthread_mutex_lock(&mtx1);
    pthread_mutex_unlock(&mtx1);
    pthread_mutex_unlock(&mtx1); // pthread double unlock
}
```

Recommendations

- Be careful not to unlock a mutex that has already been unlocked.

Links

1. Mutual exclusion
2. [pthread_mutex_unlock](#)

Vulnerability Entries

01.w_Defects/double_release.c:35

Level Medium

Status Not processed

```
32 /* pthread_mutex_lock (double_release_001_glb_mutex); */
33 double_release_001_glb_data = (double_release_001_glb_data% 100) + 1;
34 pthread_mutex_unlock (double_release_001_glb_mutex);

35 pthread_mutex_unlock (double_release_001_glb_mutex); /*Tool should detect this
line as error*/ /*ERROR:Double UnLock*/

36 return NULL;
37 }
38
```

Trace

rand ()

01.w_Defects/double_release.c:54

```
51 {  
52     while (1)  
53     {  
  
54         if (rand ())  
  
55         {  
56             double_release_001_tsk_001 (NULL);  
57         }  
58     }  
59 }
```

pthread_mutex_unlock
(double_release_001_glb_mutex); /*Tool
should detect this line as error*/ /*ERROR:
Double UnLock*/

01.w_Defects/double_release.c:35

```
32     /* pthread_mutex_lock (double_release_001_glb_mutex); */  
33     double_release_001_glb_data = (double_release_001_glb_data% 100) + 1;  
34     pthread_mutex_unlock (double_release_001_glb_mutex);  
  
35     pthread_mutex_unlock (double_release_001_glb_mutex); /*Tool should  
detect this line as error*/ /*ERROR:Double UnLock*/  
  
36     return NULL;  
37 }  
38 }
```

01.w_Defects/double_release.c:133

Level Medium**Status** Not processed

```
130     pthread_mutex_lock (double_release_003_glb_mutex);  
131     double_release_003_glb_data = (double_release_003_glb_data% 100) + 1;  
132     double_release_003_func_001 ();
```

```
133 pthread_mutex_unlock (double_release_003_glb_mutex); /*Tool should detect this  
line as error*/ /*ERROR:Double UnLock*/  
  
134 return NULL;  
135 }  
136
```

Trace

```
rand ()
```

01.w_Defects/double_release.c:151

```
148 {  
149 while (1)  
150 {  
  
151     if (rand ())  
  
152     {  
153         double_release_003_tsk_001 (NULL);  
154     }
```

```
pthread_mutex_unlock  
(double_release_003_glb_mutex); /*Tool  
should detect this line as error*/ /*ERROR:  
Double UnLock*/
```

01.w_Defects/double_release.c:133

```
130 pthread_mutex_lock (double_release_003_glb_mutex);  
131 double_release_003_glb_data = (double_release_003_glb_data% 100) + 1;  
132 double_release_003_func_001 ();  
  
133 pthread_mutex_unlock (double_release_003_glb_mutex); /*Tool should  
detect this line as error*/ /*ERROR:Double UnLock*/  
  
134 return NULL;  
135 }  
136
```

01.w_Defects/double_release.c:178

Level Medium**Status** Not processed

```
175     double_release_004_glb_data = (double_release_004_glb_data% 100) + 1;
176     pthread_mutex_unlock (double_release_004_glb_mutex);
177 }
```

178 pthread_mutex_unlock (double_release_004_glb_mutex);/*Tool should detect this line as error*/ /*ERROR:Double UnLock*/

```
179 return NULL;
180 }
181
```

Trace

rand ()

01.w_Defects/double_release.c:196

```
193 {
194     while (1)
195     {

196         if (rand ())

197         {
198             double_release_004_tsk_001 (NULL);
199         }
}
```

pthread_mutex_unlock
(double_release_004_glb_mutex);/*Tool should detect this line as error*/ /*ERROR: Double UnLock*/

01.w_Defects/double_release.c:178

```
175     double_release_004_glb_data = (double_release_004_glb_data%
100) + 1;
176     pthread_mutex_unlock (double_release_004_glb_mutex);
177 }
```

```
178 pthread_mutex_unlock (double_release_004_glb_mutex); /*Tool should  
detect this line as error*/ /*ERROR:Double UnLock*/  
  
179     return NULL;  
180 }  
181
```

01.w_Defects/double_release.c:226

Level Medium

Status Not processed

```
223 double_release_005_glb_data = (double_release_005_glb_data% 100) + 1;  
224 for(i=0;i<2;i++)  
225 {
```

```
226         pthread_mutex_unlock (double_release_005_glb_mutex); /*Tool should  
detect this line as error*/ /*ERROR:Double UnLock*/
```

```
227 }  
228 return NULL;  
229 }
```

Trace

rand ()

01.w_Defects/double_release.c:245

```
242 {  
243     while (1)  
244     {
```

```
245         if (rand ())
```

```
246     {  
247         double_release_005_tsk_001 (NULL);
```

```
248 }
```

```
pthread_mutex_unlock  
(double_release_005_glb_mutex); /*Tool  
should detect this line as error*/ /*ERROR:  
Double UnLock*/
```

01.w_Defects/double_release.c:226

```
223 double_release_005_glb_data = (double_release_005_glb_data% 100) + 1;  
224 for(i=0;i<2;i++)  
225 {  
  
226     pthread_mutex_unlock (double_release_005_glb_mutex); /*Tool  
should detect this line as error*/ /*ERROR:Double UnLock*/  
  
227 }  
228 return NULL;  
229 }
```

01.w_Defects/double_release.c:283

Level Medium

Status Not processed

```
280  
281 if(rand())  
282     pthread_mutex_unlock (double_release_006_glb_mutex);  
  
283     pthread_mutex_unlock (double_release_006_glb_mutex);/*Tool should detect  
this line as error*/ /*ERROR:Double UnLock*/  
  
284     pthread_mutex_destroy (double_release_006_glb_mutex);  
285 }  
286
```

Trace

vflag == 1

01.w_Defects/double_release.c:305

```
302 extern volatile int vflag;
303 void double_release_main ()
304 {
305     if (vflag == 1 || vflag == 888)
306     {
307         double_release_001();
308     }
```

pthread_mutex_unlock
(double_release_006_glb_mutex);/*Tool
should detect this line as error*/ /*ERROR:
Double UnLock*/

01.w_Defects/double_release.c:283

```
280
281     if(rand())
282         pthread_mutex_unlock (double_release_006_glb_mutex);
283     pthread_mutex_unlock (double_release_006_glb_mutex);/*Tool should
detect this line as error*/ /*ERROR:Double UnLock*/
284     pthread_mutex_destroy (double_release_006_glb_mutex);
285 }
286
```

01.w_Defects/unlock_without_lock.c:110

Level Medium

Status Not processed

```
107     unsigned long ip = (unsigned long)pthread_self();
108     printf("Task2! Unlock without Lock, threadID # %lu! gbl2 = %lu \n", ip,
unlock_without_lock_002_glb_data);
109 #endif /* defined(PRINT_DEBUG) */
```

```
110     pthread_mutex_unlock(&unlock_without_lock_002_glb_mutex); /*Tool should  
detect this line as error*/ /* ERROR:UnLock without lock */  
  
111 }  
112 #endif /* defined(CHECKER_POLYSPACE) */  
113 return NULL;
```

Trace

```
unlock_without_lock_002_var == (intptr_t)  
pram
```

01.w_Defects/unlock_without_lock.c:96

```
93 void * unlock_without_lock_002_tsk_001 (void * pram)  
94 {  
95 #if !defined(CHECKER_POLYSPACE)  
  
96     if(unlock_without_lock_002_var == (intptr_t)pram)  
  
97     {  
98         pthread_mutex_lock(&unlock_without_lock_002_glb_mutex);  
99         unlock_without_lock_002_glb_data = (unlock_without_lock_002_glb_data  
% 100) + 1;  
100    }
```

```
&unlock_without_lock_002_glb_mutex
```

01.w_Defects/unlock_without_lock.c:110

```
107     unsigned long ip = (unsigned long)pthread_self();  
108     printf("Task2! Unlock without Lock, threadID # %lu! gbl2 = %lu \n",ip ,  
unlock_without_lock_002_glb_data);  
109 #endif /* defined(PRINT_DEBUG) */  
  
110     pthread_mutex_unlock(&unlock_without_lock_002_glb_mutex); /*Tool  
should detect this line as error*/ /* ERROR:UnLock without lock */  
  
111 }  
112 #endif /* defined(CHECKER_POLYSPACE) */  
113 return NULL;
```

01.w_Defects/unlock_without_lock.c:202

Level Medium**Status** Not processed

```
199 unlock_without_lock_003_func_001(pram);
200 if(unlock_without_lock_003_func_002(10) > 1)
201 {
202     pthread_mutex_unlock(&unlock_without_lock_003_glb_mutex);
203 }
204 #endif /* defined(CHECKER_POLYSPACE) */
205 return NULL;
```

Trace

&unlock_without_lock_003_glb_mutex

01.w_Defects/unlock_without_lock.c:202

```
199 unlock_without_lock_003_func_001(pram);
200 if(unlock_without_lock_003_func_002(10) > 1)
201 {
202     pthread_mutex_unlock(&unlock_without_lock_003_glb_mutex);
203 }
204 #endif /* defined(CHECKER_POLYSPACE) */
205 return NULL;
```

&unlock_without_lock_003_glb_mutex

01.w_Defects/unlock_without_lock.c:202

```
199 unlock_without_lock_003_func_001(pram);
200 if(unlock_without_lock_003_func_002(10) > 1)
201 {
202     pthread_mutex_unlock(&unlock_without_lock_003_glb_mutex);
203 }
```

```
204 #endif /* defined(CHECKER_POLYSPACE) */  
205 return NULL;
```

01.w_Defects/unlock_without_lock.c:281

Level Medium

Status Not processed

```
278     if (i != 1)  
279         pthread_mutex_lock( &unlock_without_lock_004_glb_mutex_2 ); /*Tool  
should detect this line as error*/ /* ERROR:UnLock without lock */  
280     lock_never_unlock_004_glb_var += 5;  
  
281     pthread_mutex_unlock( &unlock_without_lock_004_glb_mutex_2 );  
  
282 }  
283 i--;  
284 }
```

Trace

ip >= 0

01.w_Defects/unlock_without_lock.c:276

```
273 ip = ip *20;  
274 while (i>0)  
275 {  
  
276     if (ip >= 0)  
  
277     {  
278         if (i != 1)  
279             pthread_mutex_lock( &unlock_without_lock_004_glb_mutex_2 );  
/*Tool should detect this line as error*/ /* ERROR:UnLock without lock */
```

&unlock_without_lock_004_glb_mutex_2

01.w_Defects/unlock_without_lock.c:281

```
278     if (i != 1)
279         pthread_mutex_lock( &unlock_without_lock_004_glb_mutex_2 );
/*Tool should detect this line as error*/ /* ERROR:UnLock without lock */
280     lock_never_unlock_004_glb_var += 5;

281     pthread_mutex_unlock( &unlock_without_lock_004_glb_mutex_2 );

282 }
283 i--;
284 }
```

01.w_Defects/unlock_without_lock.c:374

Level Medium**Status** Not processed

```
371     if(i !=3)
372         pthread_mutex_lock(
&unlock_without_lock_006_glb_mutex_1 );/*Tool should detect this line as error*/ /* 
ERROR:UnLock without lock */
373     unlock_without_lock_006_glb_buf[i] = 'a';

374     pthread_mutex_unlock( &unlock_without_lock_006_glb_mutex_1 );

375 }
376 }
377 #if defined PRINT_DEBUG
```

Trace

```
ip >= 0
```

01.w_Defects/unlock_without_lock.c:369

```
366 ip = ip *10;
367 for (i=0;i<5;i++)
368 {
369   if (ip >= 0)
370   {
371     if(i !=3)
372       pthread_mutex_lock(
&unlock_without_lock_006_glb_mutex_1 );/*Tool should detect this line as error*/
/* ERROR:UnLock without lock */
```

```
&unlock_without_lock_006_glb_mutex_1
```

01.w_Defects/unlock_without_lock.c:374

```
371   if(i !=3)
372     pthread_mutex_lock(
&unlock_without_lock_006_glb_mutex_1 );/*Tool should detect this line as error*/
/* ERROR:UnLock without lock */
373   unlock_without_lock_006_glb_buf[i] = 'a';

374   pthread_mutex_unlock(
&unlock_without_lock_006_glb_mutex_1 );

375 }
376 }
377 #if defined PRINT_DEBUG
```

01.w_Defects/unlock_without_lock.c:544

Level Medium

Status Not processed

```
541   unlock_without_lock_008_glb_var += 20;
542   pthread_mutex_unlock( &unlock_without_lock_008_glb_mutex_2
);
543 }
```

```
544     pthread_mutex_unlock( &unlock_without_lock_008_glb_mutex_2 );/*Tool  
should detect this line as error*/ /* ERROR:Unlock without Lock */  
  
545 }  
546 i--;  
547 } while (i>0);
```

Trace

```
ip >= 0
```

01.w_Defects/unlock_without_lock.c:536

```
533 ip = ip *20;  
534 do  
535 {  
  
536 if (ip >= 0)  
  
537 {  
538     if(i!=5)  
539     {
```

```
&unlock_without_lock_008_glb_mutex_2
```

01.w_Defects/unlock_without_lock.c:544

```
541             unlock_without_lock_008_glb_var += 20;  
542             pthread_mutex_unlock(  
&unlock_without_lock_008_glb_mutex_2 );  
543         }  
  
544     pthread_mutex_unlock( &unlock_without_lock_008_glb_mutex_2 );  
/*Tool should detect this line as error*/ /* ERROR:Unlock without Lock */  
  
545 }  
546 i--;  
547 } while (i>0);
```

02.wo_Defects/livelock.c:33

Level Medium**Status** Not processed

```
30 x=x+1;
31 pthread_mutex_unlock(&livelock_001_glb_A);
32 int status(pthread_mutex_trylock(&livelock_001_glb_B);/*Tool should not detect this
line as error*/ /*No ERROR: Live lock*/
33 pthread_mutex_unlock(&livelock_001_glb_B);

34
35 if(status==0)
36 {
```

Trace

&livelock_001_glb_B

02.wo_Defects/livelock.c:33

```
30 x=x+1;
31 pthread_mutex_unlock(&livelock_001_glb_A);
32 int status(pthread_mutex_trylock(&livelock_001_glb_B);/*Tool should not
detect this line as error*/ /*No ERROR: Live lock*/
33 pthread_mutex_unlock(&livelock_001_glb_B);

34
35 if(status==0)
36 {
```

&livelock_001_glb_B

02.wo_Defects/livelock.c:33

```
30 x=x+1;
31 pthread_mutex_unlock(&livelock_001_glb_A);
32 int status(pthread_mutex_trylock(&livelock_001_glb_B);/*Tool should not
detect this line as error*/ /*No ERROR: Live lock*/
33 pthread_mutex_unlock(&livelock_001_glb_B);
```

```
34
35 if(status==0)
36 {
```

02.wo_Defects/livelock.c:52

Level Medium

Status Not processed

```
49 pthread_mutex_unlock(&livelock_001_glb_B);
50
51 int status(pthread_mutex_trylock(&livelock_001_glb_A);

52 pthread_mutex_unlock(&livelock_001_glb_A);

53 if(status==0)
54 {
55     break;
```

Trace

&livelock_001_glb_A

02.wo_Defects/livelock.c:52

```
49 pthread_mutex_unlock(&livelock_001_glb_B);
50
51 int status(pthread_mutex_trylock(&livelock_001_glb_A);

52 pthread_mutex_unlock(&livelock_001_glb_A);

53 if(status==0)
54 {
55     break;
```

&deadlock_001_glb_A

02.wo_Defects/livelock.c:52

```
49 pthread_mutex_unlock(&deadlock_001_glb_B);
50
51 int status(pthread_mutex_trylock(&deadlock_001_glb_A));
52 pthread_mutex_unlock(&deadlock_001_glb_A);

53 if(status==0)
54 {
55     break;
```

Null pointer dereference (C/C++)

Description

Null pointer dereference may occur. This may lead to incorrect behavior of the application. The null pointer dereferencing is an operation with undefined behavior. For the implementation there are no restrictions: for example, a memory access that was not intended for the use of this program can happen (that is, when reading will be read "garbage", and when writing a value will be written to the memory area does not belong to the program).

Example

In the following example, it is assumed that the system always has a property named "cmd" defined. But if an attacker can control the program's environment so that "cmd" is not defined, the program throws a null pointer exception when it attempts to call the trim() method:

```
String cmd = System.getProperty("cmd");
cmd = cmd.trim();
```

Recommendations

- Before dereferencing a pointer, check it for equality to NULL.

Links

1. Null Dereference - OWASP
2. CWE-476: NULL Pointer Dereference

Vulnerability Entries

01.w_Defects/buffer_underrun_dynamic.c:155

Level Medium

Status Not processed

```
152 {  
153   for(j=0;j<5;j++)  
154   {  
  
155     *(*(buf+i)+j)=i; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
156   }  
157   if(i>0)  
158   free(buf[i]);
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {  
  
793   if (vflag == 1 || vflag ==888)  
  
794   {  
795     dynamic_buffer_underrun_001();  
796   }
```

```
*(*(buf+i)+j)=i
```

01.w_Defects/buffer_underrun_dynamic.c:155

```
152 {  
153   for(j=0;j<5;j++)  
154   {  
  
155     *(*(buf+i)+j)=i; /*Tool should detect this line as error*/ /*ERROR:  
Buffer Underrun*/  
  
156   }  
157   if(i>0)  
158   free(buf[i]);
```

01.w_Defects/free_nondynamically_allocated_memory.c:82

Level Medium

Status Not processed

```
79 }  
80 for(i=0;i<2;i++)  
81 {  
  
82   *((*pbuf[i])+j)=5.0;  
  
83 }  
84 free(buf1);  
85 free(buf2);
```

Trace

```
vflag == 1
```

01.w_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;  
277 void free_nondynamic_allocated_memory_main ()  
278 {
```

```
279 if (vflag == 1 || vflag ==888)

280 {
281     free_nondynamic_allocated_memory_001();
282 }
```

*((*pbuf[i])+j)=5.0

01.w_Defects/free_nondynamically_allocated_memory.c:82

```
79 }
80 for(i=0;i<2;i++)
81 {

82     *((*pbuf[i])+j)=5.0;

83 }
84 free(buf1);
85 free(buf2);
```

01.w_Defects/free_null_pointer.c:275

Level Medium

Status Not processed

```
272 free_null_pointer_009_func_001();
273 for(i=0;i<5;i++)
274 {

275     strcpy (free_null_pointer_009dst[i],"STRING");

276 }
277 while(1)
278 {
```

Trace

```
vflag == 1
```

01.w_Defects/free_null_pointer.c:570

```
567 extern volatile int vflag;
568 void free_null_pointer_main ()
569 {
570     if (vflag == 1 || vflag == 888)
571     {
572         free_null_pointer_001();
573     }
}
```

```
strcpy (free_null_pointer_009dst[i],""
STRING")
```

01.w_Defects/free_null_pointer.c:275

```
272     free_null_pointer_009_func_001();
273     for(i=0;i<5;i++)
274     {
275         strcpy (free_null_pointer_009dst[i],"STRING");
276     }
277 while(1)
278 {
```

01.w_Defects/free_null_pointer.c:452

Level Medium

Status Not processed

```
449 if (free_null_pointer_012_func_001(0) == ZERO && MAX == 1)
450 {
451     if(flag == 10)
452     a = *(ptr+1);
```

```
453 }  
454  
455 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
```

Trace

```
vflag == 1
```

01.w_Defects/free_null_pointer.c:570

```
567 extern volatile int vflag;  
568 void free_null_pointer_main ()  
569 {  
  
570 if (vflag == 1 || vflag ==888)  
  
571 {  
572     free_null_pointer_001();  
573 }
```

```
*(ptr+1)
```

01.w_Defects/free_null_pointer.c:452

```
449 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)  
450 {  
451 if(flag == 10)  
  
452 a = *(ptr+1);  
  
453 }  
454  
455 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
```

01.w_Defects/free_null_pointer.c:483

Level Medium

Status Not processed

```
480 {  
481     fptr = (double *)calloc(10, sizeof(double));  
482 }  
  
483 *(fptr+3) = 50.5;  
  
484 *fp1 = fptr;  
485 i++;  
486 }while(i>=0 && i<=1);
```

Trace

```
vflag == 1
```

01.w_Defects/free_null_pointer.c:570

```
567 extern volatile int vflag;  
568 void free_null_pointer_main ()  
569 {  
  
570 if (vflag == 1 || vflag ==888)  
  
571 {  
572     free_null_pointer_001();  
573 }
```

```
*(fptr+3) = 50.5
```

01.w_Defects/free_null_pointer.c:483

```
480 {  
481     fptr = (double *)calloc(10, sizeof(double));  
482 }  
  
483 *(fptr+3) = 50.5;  
  
484 *fp1 = fptr;  
485 i++;  
486 }while(i>=0 && i<=1);
```

01.w_Defects/func_pointer.c:177

Level Medium**Status** Not processed

```
174 {  
175   for(j=0;j<10;j++)  
176 {  
  
177       doubleptr[i][j] += 1;  
  
178 }  
179 free (doubleptr[i]);  
180 doubleptr[i] = NULL;
```

Trace

vflag == 1

01.w_Defects/func_pointer.c:617

```
614 extern volatile int vflag;  
615 void func_pointer_main ()  
616 {  
  
617   if (vflag == 1 || vflag ==888)  
  
618   {  
619       func_pointer_001();  
620   }
```

doubleptr[i][j] += 1

01.w_Defects/func_pointer.c:177

```
174 {  
175   for(j=0;j<10;j++)  
176 {  
  
177       doubleptr[i][j] += 1;  
  
178 }
```

```
179 free (doubleptr[i]);  
180 doubleptr[i] = NULL;
```

01.w_Defects/func_pointer.c:352

Level Medium

Status Not processed

```
349 func_pointer_009_u_001 *p = NULL;  
350 func_pointer_009_u_001 (*fptr)();  
351 fptr = (func_pointer_009_u_001 (*)(void))func_pointer_009_func_001;
```

352 *p = fptr();/*Tool should detect this line as error*/ /*ERROR:Bad function pointer casting*/

```
353 ret = p->b;  
354 free(p);  
355 p= NULL;
```

Trace

vflag == 1

01.w_Defects/func_pointer.c:617

```
614 extern volatile int vflag;  
615 void func_pointer_main ()  
616 {  
  
617 if (vflag == 1 || vflag ==888)  
  
618 {  
619     func_pointer_001();  
620 }
```

```
*p = fptr();/*Tool should detect this line as  
error*/ /*ERROR:Bad function pointer  
casting*/
```

01.w_Defects/func_pointer.c:352

```
349 func_pointer_009_u_001 *p = NULL;  
350 func_pointer_009_u_001 (*fptr)();  
351 fptr = (func_pointer_009_u_001 (*)(void))func_pointer_009_func_001;
```

352 *p = fptr();/*Tool should detect this line as error*/ /*ERROR:Bad function
pointer casting*/

```
353 ret = p->b;  
354 free(p);  
355 p= NULL;
```

01.w_Defects/invalid_memory_access.c:45

Level Medium

Status Not processed

```
42 }  
43 }  
44 if(flag == 10)
```

45 a = *(ptr+1); /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

```
46  
47 }  
48
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
663     if (vflag == 1 || vflag ==888)
664     {
665         invalid_memory_access_001();
666     }
```

```
*(ptr+1)
```

01.w_Defects/invalid_memory_access.c:45

```
42 }
43 }
44 if(flag == 10)

45     a = *(ptr+1); /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

46
47 }
48
```

01.w_Defects/invalid_memory_access.c:181

Level Medium

Status Not processed

```
178 }
179 for(i=0;i<5;i++)
180 {

181     *((*pbuf[i])+j)=5.0;
```

```
182 }  
183 free(buf1);  
184 free(buf2);
```

Trace

vflag == 1

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;  
661 void invalid_memory_access_main ()  
662 {  
  
663 if (vflag == 1 || vflag ==888)  
  
664 {  
665     invalid_memory_access_001();  
666 }
```

$\ast((\ast pbuf[i])+j)=5.0$

01.w_Defects/invalid_memory_access.c:181

```
178 }  
179 for(i=0;i<5;i++)  
180 {  
  
181  $\ast((\ast pbuf[i])+j)=5.0;$   
  
182 }  
183 free(buf1);  
184 free(buf2);
```

01.w_Defects/invalid_memory_access.c:265

Level Medium

Status Not processed

```
262 u->s1->a = (int *) malloc(5*sizeof(int));  
263  
264 p = u;
```

```
265 p->s1->a[0] = 1;
```

```
266  
267 free(u->s1->a);  
268 free(u->s1);
```

Trace

```
vflag == 1
```

```
01.w_Defects/invalid_memory_access.c:663
```

```
660 extern volatile int vflag;  
661 void invalid_memory_access_main ()  
662 {  
  
663     if (vflag == 1 || vflag == 888)  
  
664     {  
665         invalid_memory_access_001();  
666     }
```

```
p->s1->a[0] = 1
```

```
01.w_Defects/invalid_memory_access.c:265
```

```
262 u->s1->a = (int *) malloc(5*sizeof(int));  
263  
264 p = u;  
  
265 p->s1->a[0] = 1;  
  
266  
267 free(u->s1->a);  
268 free(u->s1);
```

01.w_Defects/invalid_memory_access.c:320

Level Medium**Status** Not processed

```
317     if(j>10)
318     break;
319 }
```

```
320     *(ptr+i) = i; /*Tool should detect this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

```
321 }
322
323 /*
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
663     if (vflag == 1 || vflag ==888)
664     {
665         invalid_memory_access_001();
666     }
}
```

```
*(ptr+i) = i
```

01.w_Defects/invalid_memory_access.c:320

```
317     if(j>10)
318     break;
319 }
```

```
320     *(ptr+i) = i; /*Tool should detect this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

```
321 }  
322  
323 /*
```

01.w_Defects/invalid_memory_access.c:371

Level Medium

Status Not processed

```
368             break;  
369         }  
370 }
```

371 return (i+s->a);/*Tool should detect this line as error*/ /*ERROR:Invalid memory access to already freed area*/

```
372 }  
373  
374 void invalid_memory_access_012 ()
```

Trace

vflag == 1

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;  
661 void invalid_memory_access_main ()  
662 {
```

663 if (vflag == 1 || vflag ==888)

```
664 {  
665     invalid_memory_access_001();  
666 }
```

```
return (i+s->a);/*Tool should detect this line  
as error*/ /*ERROR:Invalid memory access  
to already freed area*/
```

01.w_Defects/invalid_memory_access.c:371

```
368             break;  
369         }  
370 }
```

371 return (i+s->a);/*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

```
372 }  
373  
374 void invalid_memory_access_012 ()
```

01.w_Defects/invalid_memory_access.c:432

Level Medium

Status Not processed

```
429             break;  
430         }  
431 }
```

432 return invalid_memory_access_013_s_001_s_gbl->a; /*Tool should detect this line
as error*/ /*ERROR:Invalid memory access to already freed area*/

```
433 /*      return i; */  
434 }  
435
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
663     if (vflag == 1 || vflag == 888)
664     {
665         invalid_memory_access_001();
666     }
}
```

```
return
invalid_memory_access_013_s_001_s_gbl-
>a; /*Tool should detect this line as error*/
/*ERROR: Invalid memory access to already
freed area*/
```

01.w_Defects/invalid_memory_access.c:432

```
429             break;
430         }
431     }

432     return invalid_memory_access_013_s_001_s_gbl->a; /*Tool should detect
this line as error*/
/*ERROR: Invalid memory access to already freed area*/

433     /*      return i;*/
434 }
435
```

01.w_Defects/memory_allocation_failure.c:82

Level Medium

Status Not processed

```
79 {
80     for(j=0;j<5;j++)
81     {
```

```
82         *(*(ptr+i)+j)=i;
```

```
83     }
84     free(ptr[i]);
85 }
```

Trace

```
vflag == 1
```

```
01.w_Defects/memory_allocation_failure.c:724
```

```
721 extern volatile int vflag;
722 void memory_allocation_failure_main ()
723 {
724     if (vflag == 1 || vflag ==888)
725     {
726         memory_allocation_failure_001();
727     }
```

```
*(ptr+i)
```

```
01.w_Defects/memory_allocation_failure.c:82
```

```
79 {
80     for(j=0;j<5;j++)
81     {
82         *(*(ptr+i)+j)=i;
83     }
84     free(ptr[i]);
85 }
```

```
01.w_Defects/memory_leak.c:96
```

Level Medium

Status Not processed

```
93  {
94      (s+i)->buf = (char*)malloc(25* sizeof(char));/*Tool should detect this line as
error*/ /*ERROR:Memory Leakage */
95  }

96  strcpy((s+4)->buf,s1);

97  for(i= 0; i<5 ;i++);
98  free(s);
99 }
```

Trace

vflag == 1

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;
537 void memory_leak_main ()
538 {
```

```
539 if (vflag == 1 || vflag ==888)
```

```
540 {
541     memory_leak_001();
542 }
```

strcpy((s+4)->buf,s1)

01.w_Defects/memory_leak.c:96

```
93  {
94      (s+i)->buf = (char*)malloc(25* sizeof(char));/*Tool should detect this
line as error*/ /*ERROR:Memory Leakage */
95  }
```

```
96  strcpy((s+4)->buf,s1);
```

```
97  for(i= 0; i<5 ;i++);
98  free(s);
99 }
```

01.w_Defects/null_pointer.c:23

Level Medium**Status** Not processed

```
20 void null_pointer_001 ()  
21 {  
22     int *p = NULL;  
  
23     *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
dereferece*/  
  
24 }  
25  
26 /*
```

Trace

vflag == 1

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;  
354 void null_pointer_main ()  
355 {  
  
356     if (vflag == 1 || vflag == 888)  
  
357     {  
358         null_pointer_001();  
359     }
```

```
*p = 1; /*Tool should detect this line as  
error*/ /*ERROR:NULL pointer dereference*/
```

01.w_Defects/null_pointer.c:23

```
20 void null_pointer_001 ()  
21 {  
22     int *p = NULL;  
  
23     *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
derefence*/  
  
24 }  
25  
26 /*
```

01.w_Defects/null_pointer.c:34

Level Medium

Status Not processed

```
31 {  
32     int *p = NULL;  
33     int ret;  
  
34     ret = *p; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
derefence*/  
  
35     sink = ret;  
36 }  
37
```

Trace

```
vflag == 1
```

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;
354 void null_pointer_main ()
355 {
356     if (vflag == 1 || vflag == 888)
357     {
358         null_pointer_001();
359     }
}
```

```
ret = *p; /*Tool should detect this line as
error*/ /*ERROR:NULL pointer dereference*/
```

01.w_Defects/null_pointer.c:34

```
31 {
32     int *p = NULL;
33     int ret;
34     ret = *p; /*Tool should detect this line as error*/ /*ERROR:NULL pointer
derefence*/
35     sink = ret;
36 }
37
```

01.w_Defects/null_pointer.c:47

Level Medium

Status Not processed

```
44     int **pp;
45     int *p = NULL;
46     pp = &p;
```

```
47     **pp = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer
derefence*/
```

```
48 }  
49  
50 /*
```

Trace

```
vflag == 1
```

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;  
354 void null_pointer_main ()  
355 {  
  
356     if (vflag == 1 || vflag ==888)  
  
357     {  
358         null_pointer_001();  
359     }
```

```
**pp = 1
```

01.w_Defects/null_pointer.c:47

```
44     int **pp;  
45     int *p = NULL;  
46     pp = &p;  
  
47     **pp = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
dereferece*/  
  
48 }  
49  
50 /*
```

01.w_Defects/null_pointer.c:63

Level Medium

Status Not processed

```
60 void null_pointer_004 ()  
61 {  
62     null_pointer_004_s_001 *p = NULL;  
  
63     p->a = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
dereference*/  
  
64 }  
65  
66 /*
```

Trace

```
vflag == 1
```

```
01.w_Defects/null_pointer.c:356
```

```
353 extern volatile int vflag;  
354 void null_pointer_main ()  
355 {  
  
356     if (vflag == 1 || vflag == 888)  
  
357     {  
358         null_pointer_001();  
359     }
```

```
p->a = 1; /*Tool should detect this line as  
error*/ /*ERROR:NULL pointer dereference*/
```

```
01.w_Defects/null_pointer.c:63
```

```
60 void null_pointer_004 ()  
61 {  
62     null_pointer_004_s_001 *p = NULL;  
  
63     p->a = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
dereference*/  
  
64 }  
65
```

```
66 /*
```

01.w_Defects/null_pointer.c:94

Level Medium

Status Not processed

```
91 void null_pointer_005 ()  
92 {  
93     null_pointer_005_uni_001 *p = NULL;  
  
94     p->s1.a = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
dereferece*/  
  
95 }  
96  
97 /*
```

Trace

```
vflag == 1
```

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;  
354 void null_pointer_main ()  
355 {  
  
356     if (vflag == 1 || vflag == 888)  
  
357     {  
358         null_pointer_001();  
359     }
```

```
p->s1.a = 1
```

01.w_Defects/null_pointer.c:94

```
91 void null_pointer_005 ()  
92 {  
93     null_pointer_005_uni_001 *p = NULL;  
  
94     p->s1.a = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
dereference*/  
  
95 }  
96  
97 /*
```

01.w_Defects/null_pointer.c:117

Level Medium

Status Not processed

```
114 int *p;  
115 int a = 3;  
116 p = (int *)(intptr_t)((2 * a) - 6);  
  
117 *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/  
  
118 }  
119  
120 /*
```

Trace

```
vflag == 1
```

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;  
354 void null_pointer_main ()  
355 {
```

```
356 if (vflag == 1 || vflag ==888)
```

```
357 {  
358     null_pointer_001();  
359 }
```

*p = 1; /*Tool should detect this line as error*/
/*ERROR:NULL pointer dereference*/

01.w_Defects/null_pointer.c:117

```
114 int *p;  
115 int a = 3;  
116 p = (int *)(intptr_t)((2 * a) - 6);
```

117 *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/

```
118 }  
119  
120 /*
```

01.w_Defects/null_pointer.c:142

Level Medium

Status Not processed

```
139 */  
140 void null_pointer_009_func_001 (int *p)  
141 {
```

142 *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/

```
143 }  
144  
145 void null_pointer_009 ()
```

Trace

```
vflag == 1
```

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;
354 void null_pointer_main ()
355 {
356     if (vflag == 1 || vflag == 888)
357     {
358         null_pointer_001();
359     }
```

```
*p = 1; /*Tool should detect this line as error*/
/*ERROR:NULL pointer dereference*/
```

01.w_Defects/null_pointer.c:142

```
139 */
140 void null_pointer_009_func_001 (int *p)
141 {
142     *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer
dereferece*/
143 }
144
145 void null_pointer_009 ()
```

01.w_Defects/null_pointer.c:159

Level Medium

Status Not processed

```
156 int *p = NULL;
157 int *p1;
158 p1 = p;
```

```
159 *p1 = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer
dereferece*/
```

```
160 }  
161  
162 /*
```

Trace

```
vflag == 1
```

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;  
354 void null_pointer_main ()  
355 {  
  
356     if (vflag == 1 || vflag ==888)  
  
357     {  
358         null_pointer_001();  
359     }
```

*p1 = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/

01.w_Defects/null_pointer.c:159

```
156     int *p = NULL;  
157     int *p1;  
158     p1 = p;  
  
159     *p1 = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/  
  
160 }  
161  
162 /*
```

01.w_Defects/null_pointer.c:173

Level Medium

Status Not processed

```
170 int *p2;
171 p1 = p;
172 p2 = p1;
```

173 *p2 = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/

```
174 }
175
176
```

Trace

vflag == 1

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;
354 void null_pointer_main ()
355 {
```

356 if (vflag == 1 || vflag ==888)

```
357 {
358     null_pointer_001();
359 }
```

*p2 = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/

01.w_Defects/null_pointer.c:173

```
170 int *p2;
171 p1 = p;
172 p2 = p1;
```

173 *p2 = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/

```
174 }  
175  
176
```

01.w_Defects/null_pointer.c:180

Level Medium

Status Not processed

```
177 void null_pointer_012 ()  
178 {  
179   int *p = NULL;  
  
180   p[3] = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
dereference*/  
  
181 }  
182  
183 int *null_pointer_013_func_001 (void)
```

Trace

```
vflag == 1
```

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;  
354 void null_pointer_main ()  
355 {  
  
356   if (vflag == 1 || vflag == 888)  
  
357   {  
358     null_pointer_001();  
359   }
```

```
p[3] = 1; /*Tool should detect this line as  
error*/ /*ERROR:NULL pointer dereference*/
```

01.w_Defects/null_pointer.c:180

```
177 void null_pointer_012 ()  
178 {  
179   int *p = NULL;  
  
180   p[3] = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
derefence*/  
  
181 }  
182  
183 int *null_pointer_013_func_001 (void)
```

01.w_Defects/null_pointer.c:334

Level Medium

Status Not processed

```
331   null_pointer_017_func_001(0);  
332   for(i=0;i<5;i++)  
333   {  
  
334       strcpy (null_pointer_017dst[i],"STRING");/*Tool should detect this line as  
error*/ /*ERROR:NULL pointer dereference*/  
  
335   }  
336 while(1)  
337 {
```

Trace

vflag == 1

01.w_Defects/null_pointer.c:356

```
353 extern volatile int vflag;
354 void null_pointer_main ()
355 {
356     if (vflag == 1 || vflag ==888)
357     {
358         null_pointer_001();
359     }
```

strcpy (null_pointer_017dst[i],"STRING")

01.w_Defects/null_pointer.c:334

```
331     null_pointer_017_func_001(0);
332     for(i=0;i<5;i++)
333     {
334         strcpy (null_pointer_017dst[i],"STRING");/*Tool should detect this
line as error*/ /*ERROR:NULL pointer dereference*/
335     }
336 while(1)
337 {
```

01.w_Defects/uninit_memory_access.c:74

Level Medium

Status Not processed

```
71     p2 = p1;
72 }
73     ptr = &p2;
```

```
74     printf("%d \n",**ptr);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
Memory Access*/
```

```
75     free(p1);
76 }
77
```

Trace

vflag == 1

01.w_Defects/uninit_memory_access.c:462

```
459 extern volatile int vflag;
460 void uninit_memory_access_main ()
461 {
462     if (vflag == 1 || vflag == 888)
463     {
464         uninit_memory_access_001();
465     }
```

**ptr

01.w_Defects/uninit_memory_access.c:74

```
71     p2 = p1;
72 }
73     ptr = &p2;

74     printf("%d \n", **ptr); /*Tool should detect this line as error*/ /*ERROR:
Uninitialized Memory Access*/

75     free(p1);
76 }
77
```

01.w_Defects/uninit_pointer.c:131

Level Medium

Status Not processed

```
128
129 for(i=0;i<5;i++)
130 {

131 (*(*pbuf[i])+j)='a';/*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/

132 }
133     free(buf1);
134     free(buf3);
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {

421 if (vflag == 1 || vflag ==888)

422 {
423     uninit_pointer_001();
424 }
```

(*pbuf[i])

01.w_Defects/uninit_pointer.c:131

```
128
129 for(i=0;i<5;i++)
130 {

131 (*(*pbuf[i])+j)='a';/*Tool should detect this line as error*/ /*ERROR:
Uninitialized pointer*/

132 }
133     free(buf1);
134     free(buf3);
```

02.wo_Defects/buffer_underrun_dynamic.c:723

Level Medium

Status Not processed

```
720 doubleptr[i]=(char*) malloc(10*sizeof(char));/*Tool should not detect this line as  
error*/ /*No ERROR:Buffer Underrun*/  
721 if(doubleptr[i]!=NULL)  
722 {  
  
723     doubleptr[0][0]='T';  
  
724     free(doubleptr[i]);  
725 }  
726 }
```

Trace

vflag == 1

02.wo_Defects/buffer_underrun_dynamic.c:792

```
789 extern volatile int vflag;  
790 void dynamic_buffer_underrun_main ()  
791 {  
  
792     if (vflag == 1 || vflag ==888)  
  
793     {  
794         dynamic_buffer_underrun_001();  
795     }
```

doubleptr[0][0]='T'

02.wo_Defects/buffer_underrun_dynamic.c:723

```
720 doubleptr[i]=(char*) malloc(10*sizeof(char));/*Tool should not detect this line as  
error*/ /*No ERROR:Buffer Underrun*/  
721 if(doubleptr[i]!=NULL)  
722 {  
  
723     doubleptr[0][0]='T';
```

```
724 free(doubleptr[i]);  
725 }  
726 }
```

02.wo_Defects/free_nondynamically_allocated_memory.c:82

Level Medium

Status Not processed

```
79 }  
80 for(i=0;i<2;i++)  
81 {  
  
82     *((*pbuf[i])+j)=5.0;  
  
83 }  
84 free(buf1);  
85 free(buf2); /*Tool should not detect this line as error*/ /*No ERROR:Free memory not  
allocated dynamically*/
```

Trace

vflag == 1

02.wo_Defects/free_nondynamically_allocated_memory.c:279

```
276 extern volatile int vflag;  
277 void free_nondynamic_allocated_memory_main ()  
278 {  
  
279     if (vflag == 1 || vflag ==888)  
  
280     {  
281         free_nondynamic_allocated_memory_001();  
282     }
```

```
*((*pbuf[i])+j)=5.0
```

02.wo_Defects/free_nondynamically_allocated_memory.c:82

```
79 }
80 for(i=0;i<2;i++)
81 {

82     *((*pbuf[i])+j)=5.0;

83 }
84 free(buf1);
85 free(buf2); /*Tool should not detect this line as error*/ /*No ERROR:Free
memory not allocated dynamically*/
```

02.wo_Defects/invalid_memory_access.c:70

Level Medium

Status Not processed

```
67 if (staticflag == 10 && ptr!=NULL)
68         (*(ptr+1) = 10.5);
69 else

70     (*(dptr+1) = 5.5);

71
72 if(staticflag == 10 && ptr!=NULL)
73     a = *(ptr+1); /*Tool should not detect this line as error*/ /*No ERROR:Invalid
memory access to already freed area*/
```

Trace

```
vflag == 1
```

02.wo_Defects/invalid_memory_access.c:677

```
674 extern volatile int vflag;
675 void invalid_memory_access_main ()
676 {
```

```
677 if (vflag == 1 || vflag ==888)

678 {
679     invalid_memory_access_001();
680 }
```

`*(dptr+1) = 5.5`

02.wo_Defects/invalid_memory_access.c:70

```
67 if (staticflag == 10 && ptr!=NULL)
68     (*(ptr+1) = 10.5);
69 else

70     (*(dptr+1) = 5.5);

71
72 if(staticflag == 10 && ptr!=NULL)
73     a = *(ptr+1); /*Tool should not detect this line as error*/ /*No ERROR:Invalid
memory access to already freed area*/
```

02.wo_Defects/invalid_memory_access.c:183

Level Medium

Status Not processed

```
180 }
181 for(i=0;i<5;i++)
182 {

183     *((*pbuf[i])+j)=5.0;

184 }
185 if(buf2 != NULL )
186     *((*pbuf[1])+1) =buf2[0]; /*Tool should not detect this line as error*/ /*No ERROR:
Invalid memory access to already freed area*/
```

Trace

```
vflag == 1
```

02.wo_Defects/invalid_memory_access.c:677

```
674 extern volatile int vflag;
675 void invalid_memory_access_main ()
676 {
677     if (vflag == 1 || vflag ==888)
678     {
679         invalid_memory_access_001();
680     }
}
```

```
*((*pbuf[i])+j)=5.0
```

02.wo_Defects/invalid_memory_access.c:183

```
180 }
181 for(i=0;i<5;i++)
182 {

183     *((*pbuf[i])+j)=5.0;

184 }
185 if(buf2 != NULL )
186 *((*pbuf[1])+1) =buf2[0]; /*Tool should not detect this line as error*/ /*No
ERROR:Invalid memory access to already freed area*/
```

02.wo_Defects/invalid_memory_access.c:275

Level Medium

Status Not processed

```
272 u->s1->a = (int *) malloc(5*sizeof(int));
273
274 p = u;
```

```
275 p->s1->a[0] = 1; /*Tool should not detect this line as error*/ /*No ERROR:Invalid
memory access to already freed area*/
```

```
276  
277 free(u->s1->a);  
278 free(u->s1);
```

Trace

vflag == 1

02.wo_Defects/invalid_memory_access.c:677

```
674 extern volatile int vflag;  
675 void invalid_memory_access_main ()  
676 {  
  
677 if (vflag == 1 || vflag ==888)  
  
678 {  
679     invalid_memory_access_001();  
680 }
```

p->s1->a[0] = 1; /*Tool should not detect this line as error*/ /*No ERROR:Invalid memory access to already freed area*/

02.wo_Defects/invalid_memory_access.c:275

```
272 u->s1->a = (int *) malloc(5*sizeof(int));  
273  
274 p = u;
```

275 p->s1->a[0] = 1; /*Tool should not detect this line as error*/ /*No ERROR: Invalid memory access to already freed area*/

```
276  
277 free(u->s1->a);  
278 free(u->s1);
```

02.wo_Defects/memory_allocation_failure.c:84

Level Medium

Status Not processed

```
81 {  
82   for(j=0;j<5;j++)  
83   {  
  
84     *(*(ptr+i)+j)=i;  
  
85   }  
86   free(ptr[i]);  
87 }
```

Trace

```
vflag == 1
```

02.wo_Defects/memory_allocation_failure.c:741

```
738 extern volatile int vflag;  
739 void memory_allocation_failure_main ()  
740 {  
  
741   if (vflag == 1 || vflag ==888)  
  
742   {  
743     memory_allocation_failure_001 ();  
744   }
```

```
*(ptr+i)
```

02.wo_Defects/memory_allocation_failure.c:84

```
81 {  
82   for(j=0;j<5;j++)  
83   {  
  
84     *(*(ptr+i)+j)=i;  
  
85   }  
86   free(ptr[i]);  
87 }
```

02.wo_Defects/memory_leak.c:100

Level Medium**Status** Not processed

```
97 {  
98     (s+i)->buf = (char*)malloc(25* sizeof(char)); /*Tool should not detect this line as  
error*/ /*No ERROR:Memory Leakage */  
99 }  
  
100 strcpy((s+4)->buf,s1);  
  
101 for(i= 0; i<5 ;i++)  
102     free((s+i)->buf);  
103 free(s);
```

Trace

vflag == 1

02.wo_Defects/memory_leak.c:548

```
545 extern volatile int vflag;  
546 void memory_leak_main ()  
547 {  
  
548     if (vflag == 1 || vflag ==888)  
  
549     {  
550         memory_leak_001();  
551     }
```

strcpy((s+4)->buf,s1)

02.wo_Defects/memory_leak.c:100

```
97 {  
98     (s+i)->buf = (char*)malloc(25* sizeof(char)); /*Tool should not detect this line as  
error*/ /*No ERROR:Memory Leakage */  
99 }  
  
100 strcpy((s+4)->buf,s1);
```

```
101 for(i= 0; i<5 ;i++)  
102   free((s+i)->buf);  
103 free(s);
```

02.wo_Defects/uninit_memory_access.c:76

Level Medium

Status Not processed

```
73     p2 = p1;  
74 }  
75     ptr = &p2;
```

76 printf("%d \n",**ptr); /*Tool should not detect this line as error*/ /*No ERROR:
Uninitialized Memory Access*/

```
77     free(p1);  
78 }  
79
```

Trace

vflag == 1

02.wo_Defects/uninit_memory_access.c:483

```
480 extern volatile int vflag;  
481 void uninit_memory_access_main ()  
482 {
```

483 if (vflag == 1 || vflag ==888)

```
484 {  
485     uninit_memory_access_001();  
486 }
```

****ptr**

02.wo_Defects/uninit_memory_access.c:76

```
73     p2 = p1;
74 }
75     ptr = &p2;

76     printf("%d \n", **ptr); /*Tool should not detect this line as error*/ /*No
ERROR:Uninitialized Memory Access*/

77     free(p1);
78 }
79
```

Reference to array element out of bounds (C/C++)

Description

Call of the element out of array bounds may lead to incorrect application behavior, crashes, or data leak. If you erroneously call the element out of array bounds, the program reads the contents of memory cells not belonging to the array, or writes something, spoiling the contents of other variables, perhaps in other programs, and then continues its work. Specific overflows, such as overflow in a stack frame, allow an attacker to download and execute arbitrary machine code on behalf of the program and with the rights of the account from which it is executed.

Unlike memory-safe languages (e.g., Java), C does not provide built-in mechanisms to detect incorrect operations with memory.

Example

Example call of the element out of array bounds.

```
const char test1_strings_overrun(int x) {
    const char *mystr = "qwertyuiop";
    return mystr[1000];
}
```

Recommendations

- Check the correctness of the indices when working with arrays.

Links

1. CWE-129: Improper Validation of Array Index

Vulnerability Entries

01.w_Defects/buffer_overrun_dynamic.c:27

Level Medium

Status Not processed

```
24 {  
25   for (i=0;i<=5;i++)  
26   {  
  
27     buf[i]=1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
28   }  
29   free(buf);  
30 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {  
  
619   if (vflag == 1 || vflag ==888)  
  
620   {  
621     dynamic_buffer_overrun_001();  
622   }
```

buf[i]

01.w_Defects/buffer_overrun_dynamic.c:27

```
24 {  
25   for (i=0;i<=5;i++)  
26   {  
  
27     buf[i]=1; /*Tool should detect this line as error*/ /*ERROR:Buffer  
overrun*/  
  
28   }  
29   free(buf);  
30 }
```

01.w_Defects/buffer_overrun_dynamic.c:42

Level Medium**Status** Not processed

```
39 short *buf=(short*) calloc(5,sizeof(short));  
40 if(buf!=NULL)  
41 {  
  
42   *(buf+5)=1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
43   free(buf);  
44 }  
45 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {
```

```
619 if (vflag == 1 || vflag ==888)

620 {
621     dynamic_buffer_overrun_001();
622 }
```

*(buf+5)

01.w_Defects/buffer_overrun_dynamic.c:42

```
39 short *buf=(short*) calloc(5,sizeof(short));
40 if(buf!=NULL)
41 {

42     *(buf+5)=1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/

43     free(buf);
44 }
45 }
```

01.w_Defects/buffer_overrun_dynamic.c:62

Level Medium

Status Not processed

```
59 {
60     buf[i]=1;
61 }

62 ret = buf[5]; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/

63 free(buf);
64 printf("%d",ret);
65 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {

619 if (vflag == 1 || vflag ==888)

620 {
621     dynamic_buffer_overrun_001();
622 }
```

buf[5]

01.w_Defects/buffer_overrun_dynamic.c:62

```
59 {
60     buf[i]=1;
61 }

62 ret = buf[5];/*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/

63 free(buf);
64 printf("%d",ret);
65 }
```

01.w_Defects/buffer_overrun_dynamic.c:77

Level Medium

Status Not processed

```
74 int *buf=(int*) calloc(5,sizeof(int));
75 if(buf!=NULL)
76 {

77     *(buf+5) = 1;/*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
```

```
78         free(buf);
79     }
80 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
```

*(buf+5)

01.w_Defects/buffer_overrun_dynamic.c:77

```
74     int *buf=(int*) calloc(5,sizeof(int));
75     if(buf!=NULL)
76     {
77         *(buf+5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
78         free(buf);
79     }
80 }
```

01.w_Defects/buffer_overrun_dynamic.c:94

Level Medium

Status Not processed

```
91 {  
92     for(i=0;i<=5;i++)  
93     {  
  
94         buf[i]=1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
95     }  
96     free(buf);  
97 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {  
  
619     if (vflag == 1 || vflag ==888)  
  
620     {  
621         dynamic_buffer_overrun_001();  
622     }
```

buf[i]

01.w_Defects/buffer_overrun_dynamic.c:94

```
91 {  
92     for(i=0;i<=5;i++)  
93     {  
  
94         buf[i]=1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
95     }  
96     free(buf);  
97 }
```

01.w_Defects/buffer_overrun_dynamic.c:112

Level Medium**Status** Not processed

```
109 {  
110   for(i=0;i<=5;i++)  
111 {  
  
112     buf[i]=1.0; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
113 }  
114   free(buf);  
115 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {  
  
619   if (vflag == 1 || vflag ==888)  
  
620   {  
621     dynamic_buffer_overrun_001();  
622   }
```

buf[i]

01.w_Defects/buffer_overrun_dynamic.c:112

```
109 {  
110   for(i=0;i<=5;i++)  
111 {  
  
112   buf[i]=1.0; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
```

```
113 }
114     free(buf);
115 }
```

01.w_Defects/buffer_overrun_dynamic.c:130

Level Medium

Status Not processed

```
127 {
128     for(i=0;i<=5;i++)
129     {

130         buf[i]=1.0; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/

131     }
132     free(buf);
133 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {

619     if (vflag == 1 || vflag ==888)

620     {
621         dynamic_buffer_overrun_001();
622     }
```

buf[i]

01.w_Defects/buffer_overrun_dynamic.c:130

```
127 {  
128   for(i=0;i<=5;i++)  
129   {  
  
130     buf[i]=1.0; /*Tool should detect this line as error*/ /*ERROR:Buffer  
overrun*/  
  
131   }  
132   free(buf);  
133 }
```

01.w_Defects/buffer_overrun_dynamic.c:152

Level Medium**Status** Not processed

```
149 {  
150   for(j=0;j<=5;j++)  
151   {  
  
152     *((buf+i)+j)=i; /*Tool should detect this line as error*/ /*ERROR:Buffer  
overrun*/  
  
153   }  
154   free(buf[i]);  
155 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {
```

```
619 if (vflag == 1 || vflag ==888)

620 {
621     dynamic_buffer_overrun_001();
622 }
```

```
*(*(buf+i)+j)
```

01.w_Defects/buffer_overrun_dynamic.c:152

```
149 {
150 for(j=0;j<=5;j++)
151 {

152     *(*(buf+i)+j)=i; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/

153 }
154 free(buf[i]);
155 }
```

01.w_Defects/buffer_overrun_dynamic.c:174

Level Medium

Status Not processed

```
171 int i,j=6;
172 for(i=0;i<5;i++)
173 {

174     *((pbuff[i])+j)=5; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/

175 }
176 free(buf1);
177 free(buf2);
```

Trace

```
vflag == 1
```

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619   if (vflag == 1 || vflag ==888)
620   {
621     dynamic_buffer_overrun_001();
622   }
```

```
*((*pbuf[i])+j)
```

01.w_Defects/buffer_overrun_dynamic.c:174

```
171 int i,j=6;
172 for(i=0;i<5;i++)
173 {
174   *((*pbuf[i])+j)=5; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
175 }
176 free(buf1);
177 free(buf2);
```

01.w_Defects/buffer_overrun_dynamic.c:233

Level Medium

Status Not processed

```
230 int index = 5;
231 if(buf!=NULL)
232 {
```

```
233   *(buf+index)=9; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
```

```
234     free(buf);
235 }
236 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
```

*(buf+index)

01.w_Defects/buffer_overrun_dynamic.c:233

```
230 int index = 5;
231 if(buf!=NULL)
232 {
233     *(buf+index)=9; /*Tool should detect this line as error*/ /*ERROR:
Buffer overrun*/
234     free(buf);
235 }
236 }
```

01.w_Defects/buffer_overrun_dynamic.c:248

Level Medium

Status Not processed

```
245 int index = 5;
246 if(buf!=NULL)
247 {
248     buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
249     free(buf);
250 }
251 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
```

buf[index]

01.w_Defects/buffer_overrun_dynamic.c:248

```
245 int index = 5;
246 if(buf!=NULL)
247 {
248     buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
249     free(buf);
250 }
251 }
```

01.w_Defects/buffer_overrun_dynamic.c:263

Level Medium**Status** Not processed

```
260 int index = 3;
261 if(buf!=NULL)
262 {
263     *(buf +((2 * index) + 1)) = 1; /*Tool should detect this line as error*/ /*ERROR:
264     Buffer overrun*/
265 }
266 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
```

*(buf +((2 * index) + 1))

01.w_Defects/buffer_overrun_dynamic.c:263

```
260 int index = 3;
261 if(buf!=NULL)
262 {
263     *(buf +((2 * index) + 1)) = 1; /*Tool should detect this line as error*/ /*ERROR:
264     Buffer overrun*/
```

```
264     free(buf);
265 }
266 }
```

01.w_Defects/buffer_overrun_dynamic.c:278

Level Medium

Status Not processed

```
275 int index = 2;
276 if(buf!=NULL)
277 {
```

278 buf[(index * index) + 1] = 1; /*Tool should detect this line as error*/ /*ERROR:
Buffer overrun*/

```
279     free(buf);
280 }
281 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
```

619 if (vflag == 1 || vflag ==888)

```
620 {
621     dynamic_buffer_overrun_001();
622 }
```

```
buf[(index * index) + 1]
```

01.w_Defects/buffer_overrun_dynamic.c:278

```
275 int index = 2;  
276 if(buf!=NULL)  
277 {  
  
278     buf[(index * index) + 1] = 1; /*Tool should detect this line as error*/  
/*ERROR:Buffer overrun*/  
  
279     free(buf);  
280 }  
281 }
```

01.w_Defects/buffer_overrun_dynamic.c:298

Level Medium

Status Not processed

```
295 int *buf=(int*) calloc(5,sizeof(int));  
296 if(buf!=NULL)  
297 {
```

```
298     buf[dynamic_buffer_overrun_016_func_001 ()] = 1; /*Tool should detect this line  
as error*/ /*ERROR:Buffer overrun*/
```

```
299     free(buf);  
300 }  
301 }
```

Trace

```
vflag == 1
```

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {
```

```
619 if (vflag == 1 || vflag ==888)

620 {
621     dynamic_buffer_overrun_001();
622 }
```

buf[dynamic_buffer_overrun_016_func_001()
()

01.w_Defects/buffer_overrun_dynamic.c:298

```
295 int *buf=(int*) calloc(5,sizeof(int));
296 if(buf!=NULL)
297 {

298     buf[dynamic_buffer_overrun_016_func_001 ()] = 1; /*Tool should detect
this line as error*/ /*ERROR:Buffer overrun*/

299     free(buf);
300 }
301 }
```

01.w_Defects/buffer_overrun_dynamic.c:312

Level Medium

Status Not processed

```
309 int *buf=(int*) calloc(5,sizeof(int));
310 if(buf!=NULL)
311 {

312     *(buf +index) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/

313     free(buf);
314 }
315 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {

619 if (vflag == 1 || vflag ==888)

620 {
621     dynamic_buffer_overrun_001();
622 }
```

*(buf +index)

01.w_Defects/buffer_overrun_dynamic.c:312

```
309 int *buf=(int*) calloc(5,sizeof(int));
310 if(buf!=NULL)
311 {

312     *(buf +index) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/

313     free(buf);
314 }
315 }
```

01.w_Defects/buffer_overrun_dynamic.c:333

Level Medium

Status Not processed

```
330 int index = 4;
331 if(buf!=NULL)
332 {
```

```
333     *(buf+indexes[index]) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
```

```
334     free(buf);
335 }
336 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
```

indexes[index]

01.w_Defects/buffer_overrun_dynamic.c:333

```
330     int index = 4;
331     if(buf!=NULL)
332     {
333         *(buf+indexes[index]) = 1; /*Tool should detect this line as error*/
/*ERROR:Buffer overrun*/
334     free(buf);
335 }
336 }
```

01.w_Defects/buffer_overrun_dynamic.c:350

Level Medium

Status Not processed

```
347 index1 = index;
348 if(buf!=NULL)
349 {
350     buf[index1] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
351     free(buf);
352 }
353 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
}
```

buf[index1]

01.w_Defects/buffer_overrun_dynamic.c:350

```
347 index1 = index;
348 if(buf!=NULL)
349 {
350     buf[index1] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
351     free(buf);
352 }
353 }
```

01.w_Defects/buffer_overrun_dynamic.c:369

Level Medium**Status** Not processed

```
366 index2 = index1;
367 if(buf!=NULL)
368 {
369     buf[index2] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
370     free(buf);
371 }
372 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
```

buf[index2]

01.w_Defects/buffer_overrun_dynamic.c:369

```
366 index2 = index1;
367 if(buf!=NULL)
368 {
369     buf[index2] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
```

```
370     free(buf);
371 }
372 }
```

01.w_Defects/buffer_overrun_dynamic.c:387

Level Medium

Status Not processed

```
384 {
385     p1 = buf;
386     p2 = p1;

387     *(p2+5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/

388     free(buf);
389 }
390 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {

619     if (vflag == 1 || vflag == 888)

620     {
621         dynamic_buffer_overrun_001();
622     }
```

*(p2+5)

01.w_Defects/buffer_overrun_dynamic.c:387

```
384 {  
385     p1 = buf;  
386     p2 = p1;  
  
387     *(p2+5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer  
overrun*/  
  
388     free(buf);  
389 }  
390 }
```

01.w_Defects/buffer_overrun_dynamic.c:403

Level Medium

Status Not processed

```
400 if(buf!=NULL)  
401 {  
402     p = buf;  
  
403     *(p+5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
404     free(buf);  
405 }  
406 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {
```

```
619 if (vflag == 1 || vflag ==888)

620 {
621     dynamic_buffer_overrun_001();
622 }
```

*(p+5)

01.w_Defects/buffer_overrun_dynamic.c:403

```
400 if(buf!=NULL)
401 {
402     p = buf;

403     *(p+5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/

404     free(buf);
405 }
406 }
```

01.w_Defects/buffer_overrun_dynamic.c:422

Level Medium

Status Not processed

```
419 p = buf;
420 for (i = 0; i <= 5; i++)
421 {

422     *p = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/

423     p++;
424 }
425 free(buf);
```

Trace

```
vflag == 1
```

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619   if (vflag == 1 || vflag ==888)
620   {
621     dynamic_buffer_overrun_001();
622   }
```

```
*p
```

01.w_Defects/buffer_overrun_dynamic.c:422

```
419 p = buf;
420 for (i = 0; i <= 5; i++)
421 {
422   *p = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
423   p++;
424 }
425 free(buf);
```

01.w_Defects/buffer_overrun_dynamic.c:435

Level Medium

Status Not processed

```
432 */
433 void dynamic_buffer_overrun_024_func_001 (int *buf)
434 {
435   *(buf+5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
436 }
```

```
437
438 void dynamic_buffer_overrun_024 ()
```

Trace

```
vflag == 1
```

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
```

```
*(buf+5)
```

01.w_Defects/buffer_overrun_dynamic.c:435

```
432 */
433 void dynamic_buffer_overrun_024_func_001 (int *buf)
434 {
435     *(buf+5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
436 }
437
438 void dynamic_buffer_overrun_024 ()
```

01.w_Defects/buffer_overrun_dynamic.c:462

Level Medium

Status Not processed

```
459 {  
460   for(i=0;i<=5;i++)  
461   {  
  
462       buf[i]='1';/*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
463   }  
464   free(buf);  
465 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {  
  
619   if (vflag == 1 || vflag ==888)  
  
620   {  
621       dynamic_buffer_overrun_001();  
622   }
```

buf[i]

01.w_Defects/buffer_overrun_dynamic.c:462

```
459 {  
460   for(i=0;i<=5;i++)  
461   {  
  
462       buf[i]='1';/*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
463   }  
464   free(buf);  
465 }
```

01.w_Defects/buffer_overrun_dynamic.c:480

Level Medium**Status** Not processed

```
477 p = (int*)buf;
478 if(buf!=NULL)
479 {
480     *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
481     free(buf);
482 }
483 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
}
```

*(p + 5)

01.w_Defects/buffer_overrun_dynamic.c:480

```
477 p = (int*)buf;
478 if(buf!=NULL)
479 {
480     *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/
}
```

```
481     free(buf);
482 }
483 }
```

01.w_Defects/buffer_overrun_dynamic.c:496

Level Medium

Status Not processed

```
493 p = (char*)buf;
494 if(buf!=NULL)
495 {
```

496 *(p + 30) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/

```
497     free(buf);
498 }
499 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
```

619 if (vflag == 1 || vflag ==888)

```
620     {
621         dynamic_buffer_overrun_001();
622     }
```

*(p + 30)

01.w_Defects/buffer_overrun_dynamic.c:496

```
493 p = (char*)buf;
494 if(buf!=NULL)
495 {
496     *(p + 30) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
497     free(buf);
498 }
499 }
```

01.w_Defects/buffer_overrun_dynamic.c:514

Level Medium

Status Not processed

```
511     {
512     *(buf1+i)=i;
513 }
```

```
514     *(buf2+*(buf1+5))=1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
```

```
515     free(buf1);
516     free(buf2);
517 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
```

```
619 if (vflag == 1 || vflag ==888)

620 {
621     dynamic_buffer_overrun_001();
622 }
```

*(buf1+5)

01.w_Defects/buffer_overrun_dynamic.c:514

```
511 {
512     *(buf1+i)=i;
513 }

514     *(buf2+*(buf1+5))=1; /*Tool should detect this line as error*/
/*ERROR:Buffer overrun*/

515     free(buf1);
516     free(buf2);
517 }
```

01.w_Defects/buffer_overrun_dynamic.c:532

Level Medium

Status Not processed

```
529 char* buf= (char*) malloc(sizeof(char));
530 if(buf!=NULL)
531 {

532     buf[i+1]='a'; /*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/

533     free(buf);
534 }
535 i--;
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {

619 if (vflag == 1 || vflag ==888)

620 {
621     dynamic_buffer_overrun_001();
622 }
```

buf[i+1]

01.w_Defects/buffer_overrun_dynamic.c:532

```
529 char* buf= (char*) malloc(sizeof(char));
530 if(buf!=NULL)
531 {

532 buf[i+1]='a';/*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/

533 free(buf);
534 }
535 i--;
```

01.w_Defects/buffer_overrun_dynamic.c:559

Level Medium

Status Not processed

```
556 {
557 for(j=0;j<=10;j++)
558 {

559 doubleptr[i][j]='a';    /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
```

```
560 }
561 free(doubleptr[i]);
562 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {

619 if (vflag == 1 || vflag ==888)

620 {
621     dynamic_buffer_overrun_001();
622 }
```

doubleptr[i][j]

01.w_Defects/buffer_overrun_dynamic.c:559

```
556 {
557 for(j=0;j<=10;j++)
558 {

559 doubleptr[i][j]='a'; /*Tool should detect this line as error*/ /*ERROR:
Buffer overrun*/

560 }
561 free(doubleptr[i]);
562 }
```

01.w_Defects/buffer_overrun_dynamic.c:580

Level Medium

Status Not processed

```
577 {  
578     for(i=0;i<=12;i++)  
579     {  
  
580         ptr1[i]='a';/*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
581     }  
582         ptr1[11]='\0';  
583     memcpy(ptr2,ptr1,12); //vm
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {  
  
619     if (vflag == 1 || vflag ==888)  
  
620     {  
621         dynamic_buffer_overrun_001();  
622     }
```

ptr1[i]

01.w_Defects/buffer_overrun_dynamic.c:580

```
577 {  
578     for(i=0;i<=12;i++)  
579     {  
  
580         ptr1[i]='a';/*Tool should detect this line as error*/ /*ERROR:Buffer overrun*/  
  
581     }  
582         ptr1[11]='\0';  
583     memcpy(ptr2,ptr1,12); //vm
```

01.w_Defects/buffer_overrun_dynamic.c:607

Level Medium**Status** Not processed

```
604 {  
605   for(i=0;i<=10;i++)  
606   {  
  
607       ptr_s[i].arr[i]='a';/*Tool should detect this line as error*/ /*ERROR:Buffer  
overrun*//vm - Changed arri(int) to arr(char);  
  
608   }  
609   free(ptr_s);  
610 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;  
617 void dynamic_buffer_overrun_main ()  
618 {  
  
619   if (vflag == 1 || vflag ==888)  
  
620   {  
621       dynamic_buffer_overrun_001();  
622   }
```

ptr_s[i].arr[i]

01.w_Defects/buffer_overrun_dynamic.c:607

```
604 {  
605   for(i=0;i<=10;i++)  
606   {  
  
607       ptr_s[i].arr[i]='a';/*Tool should detect this line as error*/ /*ERROR:  
Buffer overrun*//vm - Changed arri(int) to arr(char);
```

```
608 }
609 free(ptr_s);
610 }
```

01.w_Defects/buffer_underrun_dynamic.c:29

Level Medium

Status Not processed

```
26 {
27 for (i=4;i>=-1;i--)
28 {
29     buf[i]=1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
30 }
31 free(buf);
32 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag ==888)
794     {
795         dynamic_buffer_underrun_001();
796     }
```

buf[i]

01.w_Defects/buffer_underrun_dynamic.c:29

```
26 {  
27 for (i=4;i>=-1;i--)  
28 {  
  
29     buf[i]=1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/  
  
30 }  
31 free(buf);  
32 }
```

01.w_Defects/buffer_underrun_dynamic.c:45

Level Medium**Status** Not processed

```
42 short *buf=(short*) calloc(5,sizeof(short));  
43 if(buf!=NULL)  
44 {  
  
45     *(buf-5)=1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/  
  
46     free(buf);  
47 }  
48 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {  
  
793     if (vflag == 1 || vflag ==888)
```

```
794 {  
795     dynamic_buffer_underrun_001();  
796 }
```

*(buf-5)

01.w_Defects/buffer_underrun_dynamic.c:45

```
42 short *buf=(short*) calloc(5,sizeof(short));  
43 if(buf!=NULL)  
44 {  
  
45     *(buf-5)=1; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
46     free(buf);  
47 }  
48 }
```

01.w_Defects/buffer_underrun_dynamic.c:65

Level Medium

Status Not processed

```
62 {  
63     buf[i]=1;  
64 }
```

65 ret = buf[-1]; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/

```
66 free(buf);  
67 printf("%d",ret);  
68 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag == 888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

buf[-1]

01.w_Defects/buffer_underrun_dynamic.c:65

```
62 {
63     buf[i] = 1;
64 }

65 ret = buf[-1]; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

66 free(buf);
67 printf("%d", ret);
68 }
```

01.w_Defects/buffer_underrun_dynamic.c:80

Level Medium

Status Not processed

```
77 int *buf = (int*) calloc(5, sizeof(int));
78 if (buf != NULL)
79 {
80     *(buf - 5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
```

```
81         free(buf);
82     }
83 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
    793 if (vflag == 1 || vflag ==888)
    794 {
    795     dynamic_buffer_underrun_001();
    796 }
```

*(buf-5)

01.w_Defects/buffer_underrun_dynamic.c:80

```
77 int *buf=(int*) calloc(5,sizeof(int));
78 if(buf!=NULL)
79 {
    80     *(buf-5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
    81     free(buf);
    82 }
83 }
```

01.w_Defects/buffer_underrun_dynamic.c:97

Level Medium

Status Not processed

```
94 {  
95   for(i=-1;i<5;i++)  
96   {  
  
97     buf[i]=1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/  
  
98   }  
99   free(buf);  
100 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {  
  
793   if (vflag == 1 || vflag ==888)  
  
794   {  
795     dynamic_buffer_underrun_001();  
796   }
```

buf[i]

01.w_Defects/buffer_underrun_dynamic.c:97

```
94 {  
95   for(i=-1;i<5;i++)  
96   {  
  
97     buf[i]=1; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
98   }  
99   free(buf);  
100 }
```

01.w_Defects/buffer_underrun_dynamic.c:115

Level Medium**Status** Not processed

```
112 {  
113   for(i=-1;i<5;i++)  
114   {  
  
115     buf[i]=1.0; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
116   }  
117   free(buf);  
118 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {  
  
793   if (vflag == 1 || vflag ==888)  
  
794   {  
795     dynamic_buffer_underrun_001();  
796   }
```

buf[i]

01.w_Defects/buffer_underrun_dynamic.c:115

```
112 {  
113   for(i=-1;i<5;i++)  
114   {  
  
115     buf[i]=1.0; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/
```

```
116 }
117     free(buf);
118 }
```

01.w_Defects/buffer_underrun_dynamic.c:133

Level Medium

Status Not processed

```
130 {
131     for(i=-1;i<5;i++)
132     {

133         buf[i]=1.0; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

134     }
135     free(buf);
136 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {

793     if (vflag == 1 || vflag ==888)

794     {
795         dynamic_buffer_underrun_001();
796     }
```

buf[i]

01.w_Defects/buffer_underrun_dynamic.c:133

```
130 {  
131   for(i=-1;i<5;i++)  
132   {  
  
133     buf[i]=1.0; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
134   }  
135   free(buf);  
136 }
```

01.w_Defects/buffer_underrun_dynamic.c:155

Level Medium**Status** Not processed

```
152 {  
153   for(j=0;j<5;j++)  
154   {  
  
155     *((buf+i)+j)=i; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
156   }  
157   if(i>0)  
158   free(buf[i]);
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {
```

```
793 if (vflag == 1 || vflag ==888)

794 {
795     dynamic_buffer_underrun_001();
796 }
```

*(buf+i)

01.w_Defects/buffer_underrun_dynamic.c:155

```
152 {
153     for(j=0;j<5;j++)
154     {

155         *(*(buf+i)+j)=i; /*Tool should detect this line as error*/ /*ERROR:
Buffer Underrun*/

156     }
157     if(i>0)
158     free(buf[i]);
```

01.w_Defects/buffer_underrun_dynamic.c:178

Level Medium

Status Not processed

```
175 int i,j=4;
176 for(i=0;i<5;i++)
177 {

178     *((pbuff[i-3])+j)=5; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

179 }
180 free(buf1);
181 free(buf2);
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag == 888)
794     {
795         dynamic_buffer_underrun_001();
796     }
```

pbuf[i-3]

01.w_Defects/buffer_underrun_dynamic.c:178

```
175 int i, j = 4;
176 for (i = 0; i < 5; i++)
177 {
178     *((*pbuf[i - 3]) + j) = 5; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
179 }
180 free(buf1);
181 free(buf2);
```

01.w_Defects/buffer_underrun_dynamic.c:237

Level Medium

Status Not processed

```
234 int index = 5;
235 if (buf != NULL)
236 {
```

```
237     *(buf - index) = 9; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
```

```
238     free(buf);
239 }
240 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag == 888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

*(buf-index)

01.w_Defects/buffer_underrun_dynamic.c:237

```
234 int index = 5;
235 if (buf != NULL)
236 {
237     *(buf + index) = 9; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
238     free(buf);
239 }
240 }
```

01.w_Defects/buffer_underrun_dynamic.c:268

Level Medium

Status Not processed

```
265 int index = 3;
266 if(buf!=NULL)
267 {
268     *(buf +((-2 * index) + 1)) = 1; /*Tool should detect this line as error*/
/*ERROR:Buffer Underrun*/
269     free(buf);
270 }
271 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag ==888)
794     {
795         dynamic_buffer_underrun_001();
796     }
```

*(buf +((-2 * index) + 1))

01.w_Defects/buffer_underrun_dynamic.c:268

```
265 int index = 3;
266 if(buf!=NULL)
267 {
268     *(buf +((-2 * index) + 1)) = 1; /*Tool should detect this line as error*/
/*ERROR:Buffer Underrun*/
269     free(buf);
270 }
271 }
```

01.w_Defects/buffer_underrun_dynamic.c:283

Level Medium**Status** Not processed

```
280 int index = 2;
281 if(buf!=NULL)
282 {
283     buf[(index * index) - 5] = 1; /*Tool should detect this line as error*/ /*ERROR:
284     Buffer Underrun*/
285     free(buf);
286 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag ==888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

buf[(index * index) - 5]

01.w_Defects/buffer_underrun_dynamic.c:283

```
280 int index = 2;
281 if(buf!=NULL)
282 {
283     buf[(index * index) - 5] = 1; /*Tool should detect this line as error*/
284     /*ERROR:Buffer Underrun*/
285 }
```

```
284     free(buf);
285 }
286 }
```

01.w_Defects/buffer_underrun_dynamic.c:303

Level Medium

Status Not processed

```
300 int *buf=(int*) calloc(5,sizeof(int));
301 if(buf!=NULL)
302 {
```

303 buf[dynamic_buffer_underrun_016_func_001 ()] = 1; /*Tool should detect this line
as error*/ /*ERROR:Buffer Underrun*/

```
304     free(buf);
305 }
306 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
```

793 if (vflag == 1 || vflag ==888)

```
794     {
795         dynamic_buffer_underrun_001();
796     }
```

```
buf  
[dynamic_buffer_underrun_016_func_001 ()]
```

01.w_Defects/buffer_underrun_dynamic.c:303

```
300 int *buf=(int*) calloc(5,sizeof(int));  
301 if(buf!=NULL)  
302 {  
  
303     buf[dynamic_buffer_underrun_016_func_001 ()] = 1; /*Tool should detect  
this line as error*/ /*ERROR:Buffer Underrun*/  
  
304     free(buf);  
305 }  
306 }
```

01.w_Defects/buffer_underrun_dynamic.c:317

Level Medium

Status Not processed

```
314 int *buf=(int*) calloc(5,sizeof(int));  
315 if(buf!=NULL)  
316 {  
  
317     *(buf -index) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
318     free(buf);  
319 }  
320 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag == 888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

*(buf -index)

01.w_Defects/buffer_underrun_dynamic.c:317

```
314     int *buf=(int*) calloc(5,sizeof(int));
315     if(buf!=NULL)
316     {
317         *(buf -index) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
318         free(buf);
319     }
320 }
```

01.w_Defects/buffer_underrun_dynamic.c:338

Level Medium

Status Not processed

```
335     int index = 2;
336     if(buf!=NULL)
337     {
```

```
338         *(buf-indexes[index]) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
```

```
339     free(buf);
340 }
341 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag == 888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

*(buf-indexes[index])

01.w_Defects/buffer_underrun_dynamic.c:338

```
335 int index = 2;
336 if(buf!=NULL)
337 {
338     *(buf-indexes[index]) = 1; /*Tool should detect this line as error*/
/*ERROR:Buffer Underrun*/
339     free(buf);
340 }
341 }
```

01.w_Defects/buffer_underrun_dynamic.c:355

Level Medium

Status Not processed

```
352 index1 = index;
353 if(buf!=NULL)
354 {
355     buf[index1] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
356     free(buf);
357 }
358 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag ==888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

buf[index1]

01.w_Defects/buffer_underrun_dynamic.c:355

```
352 index1 = index;
353 if(buf!=NULL)
354 {
355     buf[index1] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
356     free(buf);
357 }
358 }
```

01.w_Defects/buffer_underrun_dynamic.c:374

Level Medium**Status** Not processed

```
371 index2 = index1;
372 if(buf!=NULL)
373 {
374     buf[index2] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
375     free(buf);
376 }
377 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag ==888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

buf[index2]

01.w_Defects/buffer_underrun_dynamic.c:374

```
371 index2 = index1;
372 if(buf!=NULL)
373 {
374     buf[index2] = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
```

```
375     free(buf);
376 }
377 }
```

01.w_Defects/buffer_underrun_dynamic.c:392

Level Medium

Status Not processed

```
389 {
390     p1 = buf;
391     p2 = p1;

392     *(p2-5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

393     free(buf);
394 }
395 }
```

Trace

```
vflag == 1
```

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {

793     if (vflag == 1 || vflag ==888)

794     {
795         dynamic_buffer_underrun_001();
796     }
```

*(p2-5)

01.w_Defects/buffer_underrun_dynamic.c:392

```
389 {  
390     p1 = buf;  
391     p2 = p1;  
  
392     *(p2-5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
393     free(buf);  
394 }  
395 }
```

01.w_Defects/buffer_underrun_dynamic.c:408

Level Medium**Status** Not processed

```
405 if(buf!=NULL)  
406 {  
407     p = buf;  
  
408     *(p-5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/  
  
409     free(buf);  
410 }  
411 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {
```

```
793 if (vflag == 1 || vflag ==888)

794 {
795     dynamic_buffer_underrun_001();
796 }
```

*(p-5)

01.w_Defects/buffer_underrun_dynamic.c:408

```
405 if(buf!=NULL)
406 {
407     p = buf;

408     *(p-5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

409     free(buf);
410 }
411 }
```

01.w_Defects/buffer_underrun_dynamic.c:427

Level Medium

Status Not processed

```
424     p = buf;
425     for (i = 4; i >=-1; i--)
426     {

427         p[i]='1'; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/

428     }
429     free(buf);
430 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag == 888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

p[i]

01.w_Defects/buffer_underrun_dynamic.c:427

```
424     p = buf;
425     for (i = 4; i >= -1; i--)
426     {
427         p[i] = '1'; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
428     }
429     free(buf);
430 }
```

01.w_Defects/buffer_underrun_dynamic.c:439

Level Medium

Status Not processed

```
436 */
437 void dynamic_buffer_underrun_024_func_001 (int *buf)
438 {
439     *(buf - 5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
```

```
440 }
441
442 void dynamic_buffer_underrun_024 ()
```

Trace

```
vflag == 1
```

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {

793     if (vflag == 1 || vflag == 888)

794     {
795         dynamic_buffer_underrun_001();
796     }
```

```
*(buf-5)
```

01.w_Defects/buffer_underrun_dynamic.c:439

```
436 */
437 void dynamic_buffer_underrun_024_func_001 (int *buf)
438 {

439     *(buf-5) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

440 }
441
442 void dynamic_buffer_underrun_024 ()
```

01.w_Defects/buffer_underrun_dynamic.c:466

Level Medium

Status Not processed

```
463 {  
464     for(i=4;i>=-1;i--)  
465     {  
  
466         buf[i]='1'; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
467     }  
468     free(buf);  
469 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {  
  
793     if (vflag == 1 || vflag ==888)  
  
794     {  
795         dynamic_buffer_underrun_001();  
796     }
```

buf[i]

01.w_Defects/buffer_underrun_dynamic.c:466

```
463 {  
464     for(i=4;i>=-1;i--)  
465     {  
  
466         buf[i]='1'; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
467     }  
468     free(buf);  
469 }
```

01.w_Defects/buffer_underrun_dynamic.c:484

Level Medium**Status** Not processed

```
481 if(buf!=NULL)
482 {
483     p = (int*)buf;

484     *(p - 10) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/

485     free(buf);
486 }
487 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {

793     if (vflag == 1 || vflag ==888)

794     {
795         dynamic_buffer_underrun_001();
796     }
```

*(p - 10)

01.w_Defects/buffer_underrun_dynamic.c:484

```
481 if(buf!=NULL)
482 {
483     p = (int*)buf;

484     *(p - 10) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
```

```
485     free(buf);
486 }
487 }
```

01.w_Defects/buffer_underrun_dynamic.c:500

Level Medium

Status Not processed

```
497 if(buf!=NULL)
498 {
499     p = (char*)buf;

500     *(p - 10) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

501     free(buf);
502 }
503 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {

793     if (vflag == 1 || vflag ==888)

794     {
795         dynamic_buffer_underrun_001();
796     }
```

*(p - 10)

01.w_Defects/buffer_underrun_dynamic.c:500

```
497 if(buf!=NULL)
498 {
499     p = (char*)buf;
500     *(p - 10) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
501     free(buf);
502 }
503 }
```

01.w_Defects/buffer_underrun_dynamic.c:519

Level Medium

Status Not processed

```
516 {
517     *(buf1+i)=i;
518 }
```

519 *(buf2-*(buf1+4))=1; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

```
520 free(buf1);
521 free(buf2);
522 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
```

```
793 if (vflag == 1 || vflag ==888)

794 {
795     dynamic_buffer_underrun_001();
796 }
```

```
*(buf2-*(buf1+4))
```

01.w_Defects/buffer_underrun_dynamic.c:519

```
516 {
517     *(buf1+i)=i;
518 }
```

```
519 *(buf2-*(buf1+4))=1; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
```

```
520 free(buf1);
521 free(buf2);
522 }
```

01.w_Defects/buffer_underrun_dynamic.c:532

Level Medium

Status Not processed

```
529 char* buf= (char*) malloc(sizeof(char));
530 if(buf!=NULL)
531 {
```

```
532     buf[-1]='a'; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
```

```
533     free(buf);
534 }
535     break;
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag == 888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

buf[-1]

01.w_Defects/buffer_underrun_dynamic.c:532

```
529 char* buf = (char*) malloc(sizeof(char));
530 if(buf!=NULL)
531 {
532     buf[-1]='a'; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
533     free(buf);
534 }
535 break;
```

01.w_Defects/buffer_underrun_dynamic.c:559

Level Medium

Status Not processed

```
556 {
557     for(j=9;j>=-1;j--)
558     {
```

```
559     doubleptr[i][j]='a'; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
```

```
560 }
561 free(doubleptr[i]);
562 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag ==888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

doubleptr[i][j]

01.w_Defects/buffer_underrun_dynamic.c:559

```
556 {
557     for(j=9;j>=-1;j--)
558     {
559         doubleptr[i][j]='a'; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
560     }
561     free(doubleptr[i]);
562 }
```

01.w_Defects/buffer_underrun_dynamic.c:580

Level Medium

Status Not processed

```
577 ptr1[11]='\0';
578 for(i=10;i>=-1;i--) /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
579 {

580     ptr1[i]='a';

581 }
582 memcpy(ptr2,ptr1,12);
583 free(ptr1);
```

Trace

```
vflag == 1
```

```
01.w_Defects/buffer_underrun_dynamic.c:793
```

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {

793     if (vflag == 1 || vflag ==888)

794     {
795         dynamic_buffer_underrun_001();
796     }
```

```
ptr1[i]
```

```
01.w_Defects/buffer_underrun_dynamic.c:580
```

```
577 ptr1[11]='\0';
578 for(i=10;i>=-1;i--) /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
579 {

580     ptr1[i]='a';

581 }
582 memcpy(ptr2,ptr1,12);
583 free(ptr1);
```

01.w_Defects/buffer_underrun_dynamic.c:606

Level Medium**Status** Not processed

```
603 if(ptr_s!=NULL)
604 {
605     for(i=-1;i<10;i++)
606         ptr_s[i].arr[i]='a'; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
607     free(ptr_s);
608 }
609 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag ==888)
794     {
795         dynamic_buffer_underrun_001();
796     }
```

ptr_s[i].arr[i]

01.w_Defects/buffer_underrun_dynamic.c:606

```
603 if(ptr_s!=NULL)
604 {
605     for(i=-1;i<10;i++)
606         ptr_s[i].arr[i]='a'; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
```

```
607         free(ptr_s);
608 }
609 }
```

01.w_Defects/buffer_underrun_dynamic.c:621

Level Medium

Status Not processed

```
618 {
619 while(len>=-2 )
620 {

621 c = message[len];

622 if(isspace(c))
623 {
624 message[len]='\n'; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {

793 if (vflag == 1 || vflag ==888)

794 {
795     dynamic_buffer_underrun_001();
796 }
```

message[len]

01.w_Defects/buffer_underrun_dynamic.c:621

```
618 {  
619 while(len>=-2 )  
620 {  
  
621 c = message[len];  
  
622 if(isspace(c))  
623 {  
624 message[len]='\n'; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/
```

01.w_Defects/buffer_underrun_dynamic.c:624

Level Medium**Status** Not processed

```
621 c = message[len];  
622 if(isspace(c))  
623 {  
  
624 message[len]='\n'; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
625 }  
626  
627 len--;
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {
```

```
793 if (vflag == 1 || vflag ==888)

794 {
795     dynamic_buffer_underrun_001();
796 }
```

message[len]

01.w_Defects/buffer_underrun_dynamic.c:624

```
621 c = message[len];
622 if(isspace(c))
623 {

624 message[len]='\n'; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

625 }
626
627 len--;
```

01.w_Defects/buffer_underrun_dynamic.c:648

Level Medium

Status Not processed

```
645 {
646 for(i=-1;i<10;i++)
647 {

648     if(srcbuf[i]==ch) /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

649     {
650         loc=i;
651     }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag == 888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

srcbuf[i]

01.w_Defects/buffer_underrun_dynamic.c:648

```
645 {
646     for(i=-1;i<10;i++)
647     {
648         if(srcbuf[i]==ch) /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
649     }
650     loc=i;
651 }
```

01.w_Defects/buffer_underrun_dynamic.c:674

Level Medium

Status Not processed

```
671 if (loc1==0)
672 loc1--;
673
```

```
674 doubleptr[loc1][loc2]='T';
```

```
675  
676 if(loc2==0)  
677 loc2--;
```

Trace

```
vflag == 1
```

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {
```

```
793 if (vflag == 1 || vflag ==888)
```

```
794 {  
795     dynamic_buffer_underrun_001();  
796 }
```

```
doubleptr[loc1]
```

01.w_Defects/buffer_underrun_dynamic.c:674

```
671 if (loc1==0)  
672 loc1--;  
673
```

```
674 doubleptr[loc1][loc2]='T';
```

```
675  
676 if(loc2==0)  
677 loc2--;
```

01.w_Defects/buffer_underrun_dynamic.c:701

Level Medium

Status Not processed

```
698 {  
699     memcpy (newTest,test,10);  
700     char c ;  
  
701     c = newTest[-10]; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
702     free(newTest);  
703 }  
704 }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {  
  
793     if (vflag == 1 || vflag ==888)  
  
794     {  
795         dynamic_buffer_underrun_001();  
796     }
```

newTest[-10]

01.w_Defects/buffer_underrun_dynamic.c:701

```
698 {  
699     memcpy (newTest,test,10);  
700     char c ;  
  
701     c = newTest[-10]; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
702     free(newTest);  
703 }  
704 }
```

01.w_Defects/buffer_underrun_dynamic.c:721

Level Medium**Status** Not processed

```
718 {  
719   for (i=0;i<10;i++)  
720   {  
  
721       doubleptr[i-10]=(char*) malloc(10*sizeof(char)); /*Tool should detect this line  
as error*/ /*ERROR:Buffer Underrun*/  
  
722       if(doubleptr[i]!=NULL)  
723       {  
724           doubleptr[0][0]='T';
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {  
  
793   if (vflag == 1 || vflag ==888)  
  
794   {  
795       dynamic_buffer_underrun_001();  
796   }
```

doubleptr[i-10]

01.w_Defects/buffer_underrun_dynamic.c:721

```
718 {  
719   for (i=0;i<10;i++)  
720   {  
  
721       doubleptr[i-10]=(char*) malloc(10*sizeof(char)); /*Tool should detect  
this line as error*/ /*ERROR:Buffer Underrun*/
```

```
722     if(doubleptr[i]!=NULL)
723     {
724         doubleptr[0][0]='T';
```

01.w_Defects/func_pointer.c:395

Level Medium

Status Not processed

```
392 str_rev = (char *) malloc(i+1);
393 for (j = 0; j < i; j++)
394 {
395     str_rev[j] = str1[j];
396 }
397 str_rev[j] = '\0';
398 return str_rev;
```

Trace

vflag == 1

01.w_Defects/func_pointer.c:617

```
614 extern volatile int vflag;
615 void func_pointer_main ()
616 {
617     if (vflag == 1 || vflag ==888)
618     {
619         func_pointer_001();
620     }
```

str1[j]

01.w_Defects/func_pointer.c:395

```
392 str_rev = (char *) malloc(i+1);
393 for (j = 0; j < i; j++)
394 {
395     str_rev[j] = str1[j];
396 }
397 str_rev[i] = '\0';
398 return str_rev;
```

01.w_Defects/invalid_memory_access.c:320

Level Medium**Status** Not processed

```
317     if(j>10)
318         break;
319 }
```

```
320     *(ptr+i) = i; /*Tool should detect this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

```
321 }
322
323 /*
```

Trace

vflag == 1

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
```

```
663 if (vflag == 1 || vflag ==888)

664 {
665     invalid_memory_access_001();
666 }
```

*(ptr+i)

01.w_Defects/invalid_memory_access.c:320

```
317     if(j>10)
318     break;
319 }
```

```
320     *(ptr+i) = i; /*Tool should detect this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

```
321 }
322
323 /*
```

01.w_Defects/overrun_st.c:21

Level Medium

Status Not processed

```
18 void overrun_st_001 ()
19 {
20     char buf[5];

21     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

22     sink = buf[idx];
23 }
24
```

Trace

```
vflag == 1
```

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
}
```

```
buf[5]
```

01.w_Defects/overrun_st.c:21

```
18 void overrun_st_001 ()
19 {
20     char buf[5];
21
22     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
23 }
24
```

01.w_Defects/overrun_st.c:32

Level Medium

Status Not processed

```
29 void overrun_st_002 ()
30 {
31     short buf[5];
32
33     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
34 }
```

```
33     sink = buf[idx];
```

```
34 }  
35
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag ==888)  
  
785     {  
786         overrun_st_001();  
787     }
```

buf[5]

01.w_Defects/overrun_st.c:32

```
29 void overrun_st_002 ()  
30 {  
31     short buf[5];  
  
32     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
33     sink = buf[idx];  
34 }  
35
```

01.w_Defects/overrun_st.c:44

Level Medium

Status Not processed

```
41 {
```

```
42 int buf[5] = {1, 2, 3, 4, 5};  
43 int ret;  
  
44 ret = buf[5];/*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
45     sink = buf[idx];  
46 }  
47
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784 if (vflag == 1 || vflag ==888)  
  
785 {  
786     overrun_st_001();  
787 }
```

buf[5]

01.w_Defects/overrun_st.c:44

```
41 {  
42 int buf[5] = {1, 2, 3, 4, 5};  
43 int ret;  
  
44 ret = buf[5];/*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
45     sink = buf[idx];  
46 }  
47
```

01.w_Defects/overrun_st.c:55

Level Medium**Status** Not processed

```
52 void overrun_st_004 ()  
53 {  
54     int buf[5];  
  
55     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
56     sink = buf[idx];  
57 }  
58
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag == 888)  
  
785     {  
786         overrun_st_001();  
787     }
```

buf[5]

01.w_Defects/overrun_st.c:55

```
52 void overrun_st_004 ()  
53 {  
54     int buf[5];  
  
55     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
56     sink = buf[idx];
```

```
57 }  
58
```

01.w_Defects/overrun_st.c:66

Level Medium

Status Not processed

```
63 void overrun_st_005 ()  
64 {  
65     long buf[5];  
  
66     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
67     sink = buf[idx];  
68 }  
69
```

Trace

```
vflag == 1
```

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag == 888)  
  
785     {  
786         overrun_st_001();  
787     }
```

buf[5]

01.w_Defects/overrun_st.c:66

```
63 void overrun_st_005 ()  
64 {  
65     long buf[5];  
  
66     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
67     sink = buf[idx];  
68 }  
69
```

01.w_Defects/overrun_st.c:77

Level Medium**Status** Not processed

```
74 void overrun_st_006 ()  
75 {  
76     float buf[5];
```

```
77     buf[5] = 1.0; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
78     sink = buf[idx];  
79 }  
80
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {
```

```
784     if (vflag == 1 || vflag == 888)
```

```
785 {  
786     overrun_st_001();  
787 }
```

buf[5]

01.w_Defects/overrun_st.c:77

```
74 void overrun_st_006 ()  
75 {  
76     float buf[5];  
  
77     buf[5] = 1.0; /*Tool should detect this line as error*/ /*ERROR: buffer overrun  
*/  
  
78     sink = buf[idx];  
79 }  
80
```

01.w_Defects/overrun_st.c:88

Level Medium

Status Not processed

```
85 void overrun_st_007 ()  
86 {  
87     double buf[5];  
  
88     buf[5] = 1.0; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
89     sink = buf[idx];  
90 }  
91
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
```

buf[5]

01.w_Defects/overrun_st.c:88

```
85 void overrun_st_007 ()
86 {
87     double buf[5];
88     buf[5] = 1.0; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
89     sink = buf[idx];
90 }
91
```

01.w_Defects/overrun_st.c:99

Level Medium

Status Not processed

```
96 void overrun_st_008 ()
97 {
98     int buf[5][6];
99     buf[5][5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
100     sink = buf[idx][idx];
101 }
102
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
```

buf[5][5]

01.w_Defects/overrun_st.c:99

```
96 void overrun_st_008 ()
97 {
98     int buf[5][6];
99     buf[5][5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun*/
100    sink = buf[idx][idx];
101 }
102
```

01.w_Defects/overrun_st.c:110

Level Medium

Status Not processed

```
107 void overrun_st_009 ()  
108 {  
109     int buf[5][6][7];  
  
110     buf[5][5][6] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
111     sink = buf[idx][idx][idx];  
112 }  
113
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag == 888)  
  
785     {  
786         overrun_st_001();  
787     }
```

buf[5][5][6]

01.w_Defects/overrun_st.c:110

```
107 void overrun_st_009 ()  
108 {  
109     int buf[5][6][7];  
  
110     buf[5][5][6] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
111     sink = buf[idx][idx][idx];  
112 }  
113
```

01.w_Defects/overrun_st.c:126

Level Medium**Status** Not processed

```
123 int buf4[6];
124 int buf5[5];
125 int *pbuff[5] = {buf1, buf2, buf3, buf4, buf5};

126 pbuf[4][5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

127 }
128
129 /*
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {

784 if (vflag == 1 || vflag == 888)

785 {
786     overrun_st_001();
787 }
```

pbuff[4][5]

01.w_Defects/overrun_st.c:126

```
123 int buf4[6];
124 int buf5[5];
125 int *pbuff[5] = {buf1, buf2, buf3, buf4, buf5};

126 pbuff[4][5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */
```

```
127 }  
128  
129 /*
```

01.w_Defects/overrun_st.c:158

Level Medium

Status Not processed

```
155 void overrun_st_012 ()  
156 {  
157
```

158 overrun_st_012_s_gbl.buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR:
buffer overrun */

```
159 }  
160  
161 /*
```

Trace

```
vflag == 1
```

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {
```

784 if (vflag == 1 || vflag == 888)

```
785 {  
786     overrun_st_001();  
787 }
```

overrun_st_012_s_gbl.buf[5]

01.w_Defects/overrun_st.c:158

```
155 void overrun_st_012 ()  
156 {  
157  
  
158 overrun_st_012_s_gbl.buf[5] = 1; /*Tool should detect this line as error*/  
/*ERROR: buffer overrun */  
  
159 }  
160  
161 /*
```

01.w_Defects/overrun_st.c:169

Level Medium**Status** Not processed

```
166 {  
167 int buf[5];  
168 int index = 5;  
  
169 buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
170     sink = buf[idx];  
171 }  
172
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {
```

```
784 if (vflag == 1 || vflag ==888)
```

```
785 {  
786     overrun_st_001();  
787 }
```

buf[index]

01.w_Defects/overrun_st.c:169

```
166 {  
167 int buf[5];  
168 int index = 5;
```

```
169 buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer  
overrun */
```

```
170     sink = buf[idx];  
171 }  
172
```

01.w_Defects/overrun_st.c:194

Level Medium

Status Not processed

```
191 {  
192 int buf[5];  
193 int index = 2;
```

```
194 buf[(2 * index) + 1] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer  
overrun */
```

```
195     sink = buf[idx];  
196 }  
197
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
```

buf[(2 * index) + 1]

01.w_Defects/overrun_st.c:194

```
191 {
192     int buf[5];
193     int index = 2;
194     buf[(2 * index) + 1] = 1; /*Tool should detect this line as error*/ /*ERROR:
buffer overrun */
195     sink = buf[idx];
196 }
197
```

01.w_Defects/overrun_st.c:206

Level Medium

Status Not processed

```
203 {
204     int buf[5];
205     int index = 2;
```

```
206     buf[(index * index) + 1] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */
```

```
207     sink = buf[idx];
208 }
209
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
```

buf[(index * index) + 1]

01.w_Defects/overrun_st.c:206

```
203 {
204     int buf[5];
205     int index = 2;
206     buf[(index * index) + 1] = 1; /*Tool should detect this line as error*/ /*ERROR:
buffer overrun */
207     sink = buf[idx];
208 }
209
```

01.w_Defects/overrun_st.c:222

Level Medium

Status Not processed

```
219 void overrun_st_017 ()  
220 {  
221     int buf[5];  
  
222     buf[overrun_st_017_func_001()] = 1; /*Tool should detect this line as error*/  
/*ERROR: buffer overrun */  
  
223     sink = buf[idx];  
224 }  
225
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag == 888)  
  
785     {  
786         overrun_st_001();  
787     }
```

buf[overrun_st_017_func_001()]

01.w_Defects/overrun_st.c:222

```
219 void overrun_st_017 ()  
220 {  
221     int buf[5];  
  
222     buf[overrun_st_017_func_001()] = 1; /*Tool should detect this line as error*/  
/*ERROR: buffer overrun */  
  
223     sink = buf[idx];  
224 }  
225
```

01.w_Defects/overrun_st.c:233

Level Medium**Status** Not processed

```
230 int overrun_st_018_buf[5];
231 void overrun_st_018_func_001 (int index)
232 {
```

```
233    overrun_st_018_buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR:
buffer overrun */
```

```
234 }
235
236 void overrun_st_018 ()
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
```

```
784    if (vflag == 1 || vflag ==888)
```

```
785    {
786        overrun_st_001();
787    }
```

overrun_st_018_buf[index]

01.w_Defects/overrun_st.c:233

```
230 int overrun_st_018_buf[5];
231 void overrun_st_018_func_001 (int index)
232 {
```

```
233    overrun_st_018_buf[index] = 1; /*Tool should detect this line as error*/
/*ERROR: buffer overrun */
```

```
234 }  
235  
236 void overrun_st_018 ()
```

01.w_Defects/overrun_st.c:250

Level Medium

Status Not processed

```
247 int buf[5];  
248 int indexes[4] = {3, 4, 5, 6};  
249 int index = 2;
```

```
250 buf[indexes[index]] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer  
overrun */
```

```
251     sink = buf[idx];  
252 }  
253
```

Trace

```
vflag == 1
```

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {
```

```
784 if (vflag == 1 || vflag == 888)
```

```
785 {  
786     overrun_st_001();  
787 }
```

```
buf[indexes[index]]
```

01.w_Defects/overrun_st.c:250

```
247 int buf[5];
248 int indexes[4] = {3, 4, 5, 6};
249 int index = 2;
```

```
250 buf[indexes[index]] = 1; /*Tool should detect this line as error*/ /*ERROR:
buffer overrun */
```

```
251     sink = buf[idx];
252 }
253
```

01.w_Defects/overrun_st.c:264

Level Medium

Status Not processed

```
261 int index = 5;
262 int index1;
263 index1 = index;
```

```
264 buf[index1] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
265     sink = buf[idx];
266 }
267
```

Trace

```
vflag == 1
```

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
```

```
784 if (vflag == 1 || vflag ==888)
```

```
785 {  
786     overrun_st_001();  
787 }
```

buf[index1]

01.w_Defects/overrun_st.c:264

```
261 int index = 5;  
262 int index1;  
263 index1 = index;
```

```
264 buf[index1] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer  
overrun */
```

```
265     sink = buf[idx];  
266 }  
267
```

01.w_Defects/overrun_st.c:280

Level Medium

Status Not processed

```
277 int index2;  
278 index1 = index;  
279 index2 = index1;
```

```
280 buf[index2] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
281     sink = buf[idx];  
282 }  
283
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag ==888)
785     {
786         overrun_st_001();
787     }
```

buf[index2]

01.w_Defects/overrun_st.c:280

```
277 int index2;
278 index1 = index;
279 index2 = index1;

280 buf[index2] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */

281     sink = buf[idx];
282 }
283
```

01.w_Defects/overrun_st.c:293

Level Medium

Status Not processed

```
290 char buf[5];
291 char *p;
292 p = buf;
```

```
293 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
294     sink = buf[idx];
295 }
296
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
    784 if (vflag == 1 || vflag ==888)
    785 {
    786     overrun_st_001();
    787 }
```

*(p + 5)

01.w_Defects/overrun_st.c:293

```
290 char buf[5];
291 char *p;
292 p = buf;
    293 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
    294     sink = buf[idx];
    295 }
    296
```

01.w_Defects/overrun_st.c:306

Level Medium

Status Not processed

```
303 short buf[5];
304 short *p;
305 p = buf;

306 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

307     sink = buf[idx];
308 }
309
```

Trace

```
vflag == 1
```

```
01.w_Defects/overrun_st.c:784
```

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {

784 if (vflag == 1 || vflag == 888)

785 {
786     overrun_st_001();
787 }
```

```
*(p + 5)
```

```
01.w_Defects/overrun_st.c:306
```

```
303 short buf[5];
304 short *p;
305 p = buf;

306 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

307     sink = buf[idx];
308 }
309
```

01.w_Defects/overrun_st.c:320

Level Medium**Status** Not processed

```
317 int *p;
318 int ret;
319 p = buf;

320 ret = *(p + 5);/*Tool should detect this line as error*/ /*ERROR: buffer overrun */

321     sink = buf[idx];
322 }
323
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {

784 if (vflag == 1 || vflag ==888)

785 {
786     overrun_st_001();
787 }
```

*(p + 5)

01.w_Defects/overrun_st.c:320

```
317 int *p;
318 int ret;
319 p = buf;

320 ret = *(p + 5);/*Tool should detect this line as error*/ /*ERROR: buffer
overrun */
```

```
321     sink = buf[idx];
322 }
323
```

01.w_Defects/overrun_st.c:333

Level Medium

Status Not processed

```
330 int buf[5];
331 int *p;
332 p = buf;

333 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

334     sink = buf[idx];
335 }
336
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {

784 if (vflag == 1 || vflag == 888)

785 {
786     overrun_st_001();
787 }
```

*(p + 5)

01.w_Defects/overrun_st.c:333

```
330 int buf[5];
331 int *p;
332 p = buf;
```

```
333 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
*/
```

```
334     sink = buf[idx];
335 }
336
```

01.w_Defects/overrun_st.c:346

Level Medium

Status Not processed

```
343 long buf[5];
344 long *p;
345 p = buf;
```

```
346 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
347     sink = buf[idx];
348 }
349
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
```

```
784 if (vflag == 1 || vflag ==888)
```

```
785 {  
786     overrun_st_001();  
787 }
```

```
*(p + 5)
```

01.w_Defects/overrun_st.c:346

```
343 long buf[5];  
344 long *p;  
345 p = buf;
```

```
346 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
347     sink = buf[idx];  
348 }  
349
```

01.w_Defects/overrun_st.c:359

Level Medium

Status Not processed

```
356 float buf[5];  
357 float *p;  
358 p = buf;
```

```
359 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
360     sink = buf[idx];  
361 }  
362
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
```

*(p + 5)

01.w_Defects/overrun_st.c:359

```
356 float buf[5];
357 float *p;
358 p = buf;
359 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
360     sink = buf[idx];
361 }
362
```

01.w_Defects/overrun_st.c:372

Level Medium

Status Not processed

```
369 double buf[5];
370 double *p;
371 p = buf;
```

```
372 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
373     sink = buf[idx];
374 }
375
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
```

*(p + 5)

01.w_Defects/overrun_st.c:372

```
369 double buf[5];
370 double *p;
371 p = buf;
372 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun*/
373     sink = buf[idx];
374 }
375
```

01.w_Defects/overrun_st.c:387

Level Medium

Status Not processed

```
384 int **pp;
385 p = buf;
386 pp = &p;

387 /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

388     sink = buf[idx];
389 }
390
```

Trace

```
vflag == 1
```

```
01.w_Defects/overrun_st.c:784
```

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {

784 if (vflag == 1 || vflag ==888)

785 {
786     overrun_st_001();
787 }
```

```
(*pp + 5)
```

```
01.w_Defects/overrun_st.c:387
```

```
384 int **pp;
385 p = buf;
386 pp = &p;

387 /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */

388     sink = buf[idx];
389 }
390
```

01.w_Defects/overrun_st.c:402

Level Medium**Status** Not processed

```
399 int *p2;
400 p1 = buf;
401 p2 = p1;
```

```
402 p2[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
403     sink = buf[idx];
404 }
405
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
```

```
784     if (vflag == 1 || vflag == 888)
```

```
785     {
786         overrun_st_001();
787     }
```

p2[5]

01.w_Defects/overrun_st.c:402

```
399 int *p2;
400 p1 = buf;
401 p2 = p1;
```

```
402 p2[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
403     sink = buf[idx];
```

```
404 }  
405
```

01.w_Defects/overrun_st.c:415

Level Medium

Status Not processed

```
412 {  
413   int *p;  
414   p = overrun_st_031_buf_gbl;  
  
415   p[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
416 }  
417  
418 /*
```

Trace

```
vflag == 1
```

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784   if (vflag == 1 || vflag == 888)  
  
785   {  
786     overrun_st_001();  
787   }
```

p[5]

01.w_Defects/overrun_st.c:415

```
412 {  
413     int *p;  
414     p = overrun_st_031_buf_gbl;  
  
415     p[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
416 }  
417  
418 /*
```

01.w_Defects/overrun_st.c:428

Level Medium**Status** Not processed

```
425     int *p;  
426     int index = 5;  
427     p = buf;  
  
428     *(p + index) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
429     sink = buf[idx];  
430 }  
431
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {
```

```
784     if (vflag == 1 || vflag == 888)
```

```
785 {  
786     overrun_st_001();  
787 }
```

***(p + index)**

01.w_Defects/overrun_st.c:428

```
425 int *p;  
426 int index = 5;  
427 p = buf;
```

428 *(p + index) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

```
429     sink = buf[idx];  
430 }  
431
```

01.w_Defects/overrun_st.c:457

Level Medium

Status Not processed

```
454 int *p;  
455 int index = 2;  
456 p = buf;
```

457 *(p + ((2 * index) + 1)) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

```
458     sink = buf[idx];  
459 }  
460
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
}
```

*(p + ((2 * index) + 1))

01.w_Defects/overrun_st.c:457

```
454 int *p;
455 int index = 2;
456 p = buf;
457 *(p + ((2 * index) + 1)) = 1; /*Tool should detect this line as error*/ /*ERROR:
buffer overrun */
458     sink = buf[idx];
459 }
460
```

01.w_Defects/overrun_st.c:471

Level Medium

Status Not processed

```
468 int *p;
469 int index = 2;
470 p = buf;
```

471 *(p + ((index * index) + 1)) = 1; /*Tool should detect this line as error*/ /*ERROR:
buffer overrun */

```
472     sink = buf[idx];
473 }
474
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
    784 if (vflag == 1 || vflag ==888)
    785 {
        786     overrun_st_001();
    787 }
```

*(p + ((index * index) + 1))

01.w_Defects/overrun_st.c:471

```
468 int *p;
469 int index = 2;
470 p = buf;
    471 *(p + ((index * index) + 1)) = 1; /*Tool should detect this line as error*/
    /*ERROR: buffer overrun */
    472     sink = buf[idx];
    473 }
    474
```

01.w_Defects/overrun_st.c:489

Level Medium

Status Not processed

```
486 int buf[5];
487 int *p;
488 p = buf;

489 *(p + overrun_st_036_func_001()) = 1; /*Tool should detect this line as error*/
/*ERROR: buffer overrun */

490     sink = buf[idx];
491 }
492
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {

784 if (vflag == 1 || vflag == 888)

785 {
786     overrun_st_001();
787 }
```

*(p + overrun_st_036_func_001())

01.w_Defects/overrun_st.c:489

```
486 int buf[5];
487 int *p;
488 p = buf;

489 *(p + overrun_st_036_func_001()) = 1; /*Tool should detect this line as error*/
/*ERROR: buffer overrun */

490     sink = buf[idx];
491 }
492
```

01.w_Defects/overrun_st.c:502

Level Medium**Status** Not processed

```
499 int buf[5];
500 int *p;
501 p = buf;
```

```
502 *(p + index) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
503     sink = buf[idx];
504 }
505
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
```

```
784     if (vflag == 1 || vflag == 888)
```

```
785     {
786         overrun_st_001();
787     }
```

*(p + index)

01.w_Defects/overrun_st.c:502

```
499 int buf[5];
500 int *p;
501 p = buf;
```

```
502 *(p + index) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
503     sink = buf[idx];
504 }
505
```

01.w_Defects/overrun_st.c:522

Level Medium

Status Not processed

```
519 int indexes[4] = {3, 4, 5, 6};
520 int index = 2;
521 p = buf;
```

522 *(p + indexes[index]) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

```
523     sink = buf[idx];
524 }
525
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
```

784 if (vflag == 1 || vflag == 888)

```
785 {
786     overrun_st_001();
787 }
```

*(p + indexes[index])

01.w_Defects/overrun_st.c:522

```
519 int indexes[4] = {3, 4, 5, 6};  
520 int index = 2;  
521 p = buf;
```

522 *(p + indexes[index]) = 1; /*Tool should detect this line as error*/ /*ERROR:
buffer overrun */

```
523     sink = buf[idx];  
524 }  
525
```

01.w_Defects/overrun_st.c:538

Level Medium

Status Not processed

```
535 int index1;  
536 index1 = index;  
537 p = buf;
```

538 *(p + index1) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

```
539     sink = buf[idx];  
540 }  
541
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {
```

```
784 if (vflag == 1 || vflag ==888)
```

```
785 {  
786     overrun_st_001();  
787 }
```

`*(p + index1)`

01.w_Defects/overrun_st.c:538

```
535 int index1;  
536 index1 = index;  
537 p = buf;
```

```
538 *(p + index1) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer  
overrun */
```

```
539     sink = buf[idx];  
540 }  
541
```

01.w_Defects/overrun_st.c:556

Level Medium

Status Not processed

```
553 index1 = index;  
554 index2 = index1;  
555 p = buf;
```

```
556 *(p + index2) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
557     sink = buf[idx];  
558 }  
559
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
}
```

*(p + index2)

01.w_Defects/overrun_st.c:556

```
553     index1 = index;
554     index2 = index1;
555     p = buf;
556     *(p + index2) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */
557     sink = buf[idx];
558 }
559
```

01.w_Defects/overrun_st.c:570

Level Medium

Status Not processed

```
567     int i;
568     for (i = 0; i <= 5; i++)
569     {
```

```
570         buf[i] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
571 }
572     sink = buf[idx];
573 }
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
    784 if (vflag == 1 || vflag ==888)
    785 {
        786     overrun_st_001();
    787 }
```

buf[i]

01.w_Defects/overrun_st.c:570

```
567 int i;
568 for (i = 0; i <= 5; i++)
569 {
    570     buf[i] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */
    571 }
    572     sink = buf[idx];
    573 }
```

01.w_Defects/overrun_st.c:588

Level Medium

Status Not processed

```
585 {  
586   for (j = 0; j < 6; j ++)  
587   {  
  
588       buf[i][j] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun  
*/  
  
589   }  
590 }  
591   sink = buf[idx][idx];
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784   if (vflag == 1 || vflag ==888)  
  
785   {  
786       overrun_st_001();  
787   }
```

buf[i][j]

01.w_Defects/overrun_st.c:588

```
585 {  
586   for (j = 0; j < 6; j ++)  
587   {  
  
588       buf[i][j] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun  
*/  
  
589   }  
590 }  
591   sink = buf[idx][idx];
```

01.w_Defects/overrun_st.c:613

Level Medium**Status** Not processed

```
610 {  
611     for (j = 0; j < 6; j++)  
612     {  
  
613         pbuf[i][j] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer  
overrun */  
  
614     }  
615 }  
616 }
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag == 888)  
  
785     {  
786         overrun_st_001();  
787     }
```

pbuf[i][j]

01.w_Defects/overrun_st.c:613

```
610 {  
611     for (j = 0; j < 6; j++)  
612     {  
  
613         pbuf[i][j] = 1; /*Tool should detect this line as error*/ /*ERROR:  
buffer overrun */
```

```
614      }
615  }
616 }
```

01.w_Defects/overrun_st.c:630

Level Medium

Status Not processed

```
627 p = buf;
628 for (i = 0; i <= 5; i++)
629 {
630   *p = 1;
631   p ++; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
632 }
633   sink = buf[idx];
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784   if (vflag == 1 || vflag == 888)
785   {
786     overrun_st_001();
787   }
```

```
*p
```

01.w_Defects/overrun_st.c:630

```
627 p = buf;
628 for (i = 0; i <= 5; i++)
629 {
630   *p = 1;
631   p ++; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
632 }
633   sink = buf[idx];
```

01.w_Defects/overrun_st.c:642

Level Medium

Status Not processed

```
639 */
640 void overrun_st_045_func_001 (int buf[])
641 {
642   buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
643 }
644
645 void overrun_st_045 ()
```

Trace

```
vflag == 1
```

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
```

```
784   if (vflag == 1 || vflag == 888)
```

```
785 {  
786     overrun_st_001();  
787 }
```

buf[5]

01.w_Defects/overrun_st.c:642

```
639 */  
640 void overrun_st_045_func_001 (int buf[])  
641 {  
  
642 buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
643 }  
644  
645 void overrun_st_045 ()
```

01.w_Defects/overrun_st.c:658

Level Medium

Status Not processed

```
655 */  
656 void overrun_st_046_func_001 (int *p)  
657 {  
  
658 *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
659 }  
660  
661 void overrun_st_046 ()
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {

784     if (vflag == 1 || vflag ==888)

785     {
786         overrun_st_001();
787     }
```

*(p + 5)

01.w_Defects/overrun_st.c:658

```
655 */
656 void overrun_st_046_func_001 (int *p)
657 {

658     *(p + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun
 */

659 }
660
661 void overrun_st_046 ()
```

01.w_Defects/overrun_st.c:674

Level Medium

Status Not processed

```
671 */
672 void overrun_st_047_func_001 (int *p)
673 {
```

674 p[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

```
675 }  
676  
677 void overrun_st_047 ()
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag == 888)  
  
785     {  
786         overrun_st_001();  
787     }
```

p[5]

01.w_Defects/overrun_st.c:674

```
671 */  
672 void overrun_st_047_func_001 (int *p)  
673 {  
  
674     p[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
675 }  
676  
677 void overrun_st_047 ()
```

01.w_Defects/overrun_st.c:689

Level Medium

Status Not processed

```
686 */
687 void overrun_st_048_func_001 (int buf[])
688 {

689 *(buf + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

690 }
691
692 void overrun_st_048 ()
```

Trace

```
vflag == 1
```

```
01.w_Defects/overrun_st.c:784
```

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {

784 if (vflag == 1 || vflag == 888)

785 {
786     overrun_st_001();
787 }
```

```
*(buf + 5)
```

```
01.w_Defects/overrun_st.c:689
```

```
686 */
687 void overrun_st_048_func_001 (int buf[])
688 {

689 *(buf + 5) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */

690 }
691
692 void overrun_st_048 ()
```

01.w_Defects/overrun_st.c:706

Level Medium**Status** Not processed

```
703 void overrun_st_049 ()  
704 {  
705     int buf[] = {1, 2, 3, 4, 5};  
  
706     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
707     sink = buf[idx];  
708 }  
709
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag == 888)  
  
785     {  
786         overrun_st_001();  
787     }
```

buf[5]

01.w_Defects/overrun_st.c:706

```
703 void overrun_st_049 ()  
704 {  
705     int buf[] = {1, 2, 3, 4, 5};  
  
706     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
707     sink = buf[idx];
```

```
708 }  
709
```

01.w_Defects/overrun_st.c:724

Level Medium

Status Not processed

```
721             {1, 2, 3, 4, 5, 6},  
722             {1, 2, 3, 4, 5, 6}  
723         };
```

724 buf[5][5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */

```
725 }  
726  
727 /*
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {
```

784 if (vflag == 1 || vflag == 888)

```
785 {  
786     overrun_st_001();  
787 }
```

buf[5][5]

01.w_Defects/overrun_st.c:724

```
721             {1, 2, 3, 4, 5, 6},  
722             {1, 2, 3, 4, 5, 6}  
723         };
```

```
724     buf[5][5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
725 }  
726  
727 /*
```

01.w_Defects/overrun_st.c:739

Level Medium**Status** Not processed

```
736 int buf4[] = {1, 2, 3, 4, 5, 6};  
737 int buf5[] = {1, 2, 3, 4, 5};  
738 int *pbuff[] = {buf1, buf2, buf3, buf4, buf5};
```

```
739     pbuf[4][5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
740 }  
741  
742 /*
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {
```

```
784 if (vflag == 1 || vflag ==888)
```

```
785 {  
786     overrun_st_001();  
787 }
```

pbuf[4][5]

01.w_Defects/overrun_st.c:739

```
736 int buf4[] = {1, 2, 3, 4, 5, 6};  
737 int buf5[] = {1, 2, 3, 4, 5};  
738 int *pbuf[] = {buf1, buf2, buf3, buf4, buf5};
```

```
739 pbuf[4][5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer  
overrun */
```

```
740 }  
741  
742 /*
```

01.w_Defects/overrun_st.c:749

Level Medium

Status Not processed

```
746 void overrun_st_052 ()  
747 {  
748     char buf[] = "1234";
```

```
749     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
750 }  
751  
752 /*
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
}
```

buf[5]

01.w_Defects/overrun_st.c:749

```
746 void overrun_st_052 ()
747 {
748     char buf[] = "1234";
749     buf[5] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
750 }
751
752 /*
```

01.w_Defects/overrun_st.c:761

Level Medium

Status Not processed

```
758     char buf[8];
759     int *p;
760     p = (int*)buf;
```

```
761     *(p + 2) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
762 }
```

```
763  
764 /*
```

Trace

```
vflag == 1
```

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag ==888)  
  
785     {  
786         overrun_st_001();  
787     }
```

```
*(p + 2)
```

01.w_Defects/overrun_st.c:761

```
758 char buf[8];  
759 int *p;  
760 p = (int*)buf;  
  
761     *(p + 2) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun  
*/  
  
762 }  
763  
764 /*
```

01.w_Defects/overrun_st.c:773

Level Medium

Status Not processed

```
770 {  
771     char *p;  
772     p = (char*)overrun_st_054_buf_gbl;  
  
773     *(p + 50) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
774 }  
775  
776
```

Trace

```
vflag == 1
```

```
01.w_Defects/overrun_st.c:784
```

```
781 extern volatile int vflag;  
782 void overrun_st_main ()  
783 {  
  
784     if (vflag == 1 || vflag == 888)  
  
785     {  
786         overrun_st_001();  
787     }
```

```
*(p + 50)
```

```
01.w_Defects/overrun_st.c:773
```

```
770 {  
771     char *p;  
772     p = (char*)overrun_st_054_buf_gbl;  
  
773     *(p + 50) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
774 }  
775  
776
```

01.w_Defects/st_underrun.c:25

Level Medium**Status** Not processed

```
22 char buf[10];
23 strcpy(buf, "my string");
24 int len = strlen(buf) - 1;

25 while (buf[len] != 'Z')

26 {
27     len--; /*Tool should detect this line as error*/ /* Stack Under RUN error */
28     /* if (buf[len] == '\0' )
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {

263     if (vflag == 1 || vflag ==888)

264     {
265         st_underrun_001();
266     }
```

buf[len]

01.w_Defects/st_underrun.c:25

```
22 char buf[10];
23 strcpy(buf, "my string");
24 int len = strlen(buf) - 1;

25 while (buf[len] != 'Z')

26 {
```

```
27 len--; /*Tool should detect this line as error*/ /* Stack Under RUN error */
28 /* if (buf[len] == '\0' )
```

01.w_Defects/st_underrun.c:51

Level Medium**Status** Not processed

```
48 {
49
50     int len = strlen(s.buf) - 1;

51     for (;s.buf[len] != 'Z';len--)/*Tool should detect this line as error*/ /* Stack Under
RUN error */

52     {
53         /* if (s.buf[len] == '\0')
54             break; */
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {

263     if (vflag == 1 || vflag ==888)

264     {
265         st_underrun_001();
266     }
```

```
s.buf[len]
```

01.w_Defects/st_underrun.c:51

```
48 {  
49  
50     int len = strlen(s.buf) - 1;  
  
51     for (;s.buf[len] != 'Z';len--)/*Tool should detect this line as error*/ /* Stack  
Under RUN error */  
  
52     {  
53         /* if (s.buf[len] == '\0')  
54             break; */
```

01.w_Defects/st_underrun.c:88

Level Medium

Status Not processed

```
85 int len = strlen(s->buf) - 1;  
86 do  
87 {  
  
88     s->buf[len] = 'A';  
  
89     len--;  
90     /* if (s->buf[len] == '\0')  
91         break; */
```

Trace

```
vflag == 1
```

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;  
261 void st_underrun_main ()  
262 {
```

```
263 if (vflag == 1 || vflag ==888)
```

```
264 {  
265     st_underrun_001();  
266 }
```

```
s->buf[len]
```

01.w_Defects/st_underrun.c:88

```
85 int len = strlen(s->buf) - 1;  
86 do  
87 {  
  
88 s->buf[len] = 'A';  
  
89 len--;  
90 /* if (s->buf[len] == '\0')  
91     break; */
```

01.w_Defects/st_underrun.c:92

Level Medium

Status Not processed

```
89     len--;  
90     /* if (s->buf[len] == '\0')  
91         break; */
```

```
92 }while (s->buf[len] != 'Z');/*Tool should detect this line as error*/ /* Stack Under  
RUN error */
```

```
93 }  
94  
95 void st_underrun_003 ()
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {
263     if (vflag == 1 || vflag == 888)
264     {
265         st_underrun_001();
266     }
```

s->buf[len]

01.w_Defects/st_underrun.c:92

```
89         len--;
90         /* if (s->buf[len] == '\0')
91             break; */
92     }while (s->buf[len] != 'Z');/*Tool should detect this line as error*/ /* Stack
Under RUN error */
93 }
94
95 void st_underrun_003 ()
```

01.w_Defects/st_underrun.c:125

Level Medium

Status Not processed

```
122 int len = strlen(s->buf) - 1;
123 do
124 {
125     s->buf[len] = 'B';
```

```
126 s1.buf[len] = s->buf[len];  
127 len--;  
128 /* if (s->buf[len] == '\0')
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;  
261 void st_underrun_main ()  
262 {  
  
263 if (vflag == 1 || vflag == 888)  
  
264 {  
265     st_underrun_001();  
266 }
```

s->buf[len]

01.w_Defects/st_underrun.c:125

```
122 int len = strlen(s->buf) - 1;  
123 do  
124 {  
  
125 s->buf[len] = 'B';  
  
126 s1.buf[len] = s->buf[len];  
127 len--;  
128 /* if (s->buf[len] == '\0')
```

01.w_Defects/st_underrun.c:126

Level Medium

Status Not processed

```
123 do
124 {
125   s->buf[len] = 'B';

126   s1.buf[len] = s->buf[len];

127   len--;
128   /* if (s->buf[len] == '\0')
129   break; */
```

Trace

```
vflag == 1
```

```
01.w_Defects/st_underrun.c:263
```

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {

263   if (vflag == 1 || vflag == 888)

264   {
265     st_underrun_001();
266 }
```

```
s->buf[len]
```

```
01.w_Defects/st_underrun.c:126
```

```
123 do
124 {
125   s->buf[len] = 'B';

126   s1.buf[len] = s->buf[len];

127   len--;
128   /* if (s->buf[len] == '\0')
129   break; */
```

01.w_Defects/st_underrun.c:126

Level Medium**Status** Not processed

```
123 do
124 {
125   s->buf[len] = 'B';

126   s1.buf[len] = s->buf[len];

127   len--;
128   /* if (s->buf[len] == '\0')
129     break; */
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {

263   if (vflag == 1 || vflag == 888)

264   {
265     st_underrun_001();
266   }
```

s1.buf[len]

01.w_Defects/st_underrun.c:126

```
123 do
124 {
125   s->buf[len] = 'B';

126   s1.buf[len] = s->buf[len];

127   len--;
```

```
128 /* if (s->buf[len] == '\0')
129     break; */
```

01.w_Defects/st_underrun.c:130

Level Medium

Status Not processed

```
127     len--;
128     /* if (s->buf[len] == '\0'
129         break; */
```

130 }while (s->buf[len] != 'Z');/*Tool should detect this line as error*/ /* Stack Under
RUN error */

```
131 return s1;
132 }
133
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {
```

263 if (vflag == 1 || vflag ==888)

```
264 {
265     st_underrun_001();
266 }
```

```
s->buf[len]
```

01.w_Defects/st_underrun.c:130

```
127     len--;
128     /* if (s->buf[len] == '\0')
129         break; */

130 }while (s->buf[len] != 'Z');/*Tool should detect this line as error*/ /* Stack
Under RUN error */

131 return s1;
132 }
133
```

01.w_Defects/st_underrun.c:150

Level Medium

Status Not processed

```
147
148 void st_underrun_005_func_001 (st_underrun_005_s_001 s, int cnt)
149 {

150 while (s.buf[cnt] != 'Z' )

151 {
152     cnt--;
153     if(cnt>0)
```

Trace

```
vflag == 1
```

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {
```

```
263 if (vflag == 1 || vflag ==888)
```

```
264 {  
265     st_underrun_001();  
266 }
```

s.buf[cnt]

01.w_Defects/st_underrun.c:150

```
147  
148 void st_underrun_005_func_001 (st_underrun_005_s_001 s, int cnt)  
149 {
```

```
150 while (s.buf[cnt] != 'Z' )
```

```
151 {  
152     cnt--;  
153     if(cnt>0)
```

01.w_Defects/st_underrun.c:160

Level Medium

Status Not processed

```
157 else  
158 {  
159     /*break;*/ /*Tool should detect this line as error*/ /* Stack Under RUN error  
*/
```

```
160         s.buf[cnt] = 'C';
```

```
161     }  
162 }  
163 }
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {
263     if (vflag == 1 || vflag == 888)
264     {
265         st_underrun_001();
266     }
```

s.buf[cnt]

01.w_Defects/st_underrun.c:160

```
157     else
158     {
159         /*break; */ /*Tool should detect this line as error*/ /* Stack Under RUN
error */
160         s.buf[cnt] = 'C';
161     }
162 }
163 }
```

01.w_Defects/st_underrun.c:193

Level Medium

Status Not processed

```
190
191 int len = strlen(s.buf) - 1;
192 char c;
```

```
193 for (; s.buf[len] != 'Z'; len--) /*Tool should detect this line as error*/ /* Error: Stack
Under RUN error */
```

```
194 {  
195     c = s.buf[len];  
196 /*if (0)
```

Trace

s.buf[len]

01.w_Defects/st_underrun.c:193

```
190  
191 int len = strlen(s.buf) - 1;  
192 char c;
```

193 for (;s.buf[len] != 'Z';len--) /*Tool should detect this line as error*/ /* Error:
Stack Under RUN error */

```
194 {  
195     c = s.buf[len];  
196 /*if (0)
```

s.buf[len]

01.w_Defects/st_underrun.c:193

```
190  
191 int len = strlen(s.buf) - 1;  
192 char c;
```

193 for (;s.buf[len] != 'Z';len--) /*Tool should detect this line as error*/ /* Error:
Stack Under RUN error */

```
194 {  
195     c = s.buf[len];  
196 /*if (0)
```

01.w_Defects/st_underrun.c:195

Level Medium

Status Not processed

```
192 char c;
193 for (;s.buf[len] != 'Z';len--) /*Tool should detect this line as error*/ /* Error: Stack
Under RUN error */
194 {

195     c = s.buf[len];

196 /*if (0)
197 break;*/
198 }
```

Trace

```
s.buf[len] != 'Z'
```

```
01.w_Defects/st_underrun.c:193
```

```
190
191 int len = strlen(s.buf) - 1;
192 char c;
```

```
193 for (;s.buf[len] != 'Z';len--) /*Tool should detect this line as error*/ /* Error:
Stack Under RUN error */
```

```
194 {
195     c = s.buf[len];
196 /*if (0)
```

```
s.buf[len]
```

```
01.w_Defects/st_underrun.c:195
```

```
192 char c;
193 for (;s.buf[len] != 'Z';len--) /*Tool should detect this line as error*/ /* Error:
Stack Under RUN error */
194 {
```

```
195     c = s.buf[len];
```

```
196 /*if (0)
197 break;*/
198 }
```

01.w_Defects/st_underrun.c:227

Level Medium**Status** Not processed

```
224 {  
225   int len = strlen(s->buf) - 1;  
226   char c;  
  
227   for (;s->buf[len] != 'Z';len--)  
  
228   {  
229     c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under RUN error  
*/  
230     /*if (s->buf[len] == '\0')
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;  
261 void st_underrun_main ()  
262 {  
  
263   if (vflag == 1 || vflag ==888)  
  
264   {  
265     st_underrun_001();  
266   }
```

s->buf[len]

01.w_Defects/st_underrun.c:227

```
224 {  
225   int len = strlen(s->buf) - 1;  
226   char c;  
  
227   for (;s->buf[len] != 'Z';len--)
```

```
228  {
229      c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under
RUN error */
230          /*if (s->buf[len] == '\0')
```

01.w_Defects/st_underrun.c:229

Level Medium**Status** Not processed

```
226 char c;
227 for (;s->buf[len] != 'Z';len--)
228 {

229     c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under RUN error
*/
230     /*if (s->buf[len] == '\0')
231     break;*/
232 }
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {

263     if (vflag == 1 || vflag ==888)

264     {
265         st_underrun_001();
266     }
```

```
s->buf[len]
```

01.w_Defects/st_underrun.c:229

```
226 char c;
227 for (;s->buf[len] != 'Z';len--)
228 {

229     c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under
RUN error */

230 /*if (s->buf[len] == '\0')
231     break;*/
232 }
```

01.w_Defects/underrun_st.c:21

Level Medium

Status Not processed

```
18 {
19     int buf[5] = {1, 2, 3, 4, 5};
20     int ret;

21     ret = buf[-1];/*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

22 }
23
24 /*
```

Trace

```
vflag == 1
```

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {
```

```
202 if (vflag == 1 || vflag ==888)
```

```
203 {  
204     underrun_st_001();  
205 }
```

buf[-1]

01.w_Defects/underrun_st.c:21

```
18 {  
19     int buf[5] = {1, 2, 3, 4, 5};  
20     int ret;  
  
21     ret = buf[-1];/*Tool should detect this line as error*/ /*ERROR:Data  
Underrun*/  
  
22 }  
23  
24 /*
```

01.w_Defects/underrun_st.c:31

Level Medium

Status Not processed

```
28 void underrun_st_002 ()  
29 {  
30     int buf[5];  
  
31     buf[-1] = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/  
  
32 }  
33  
34 /*
```

Trace

```
vflag == 1
```

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {

202 if (vflag == 1 || vflag ==888)

203 {
204     underrun_st_001();
205 }
```

```
buf[-1]
```

01.w_Defects/underrun_st.c:31

```
28 void underrun_st_002 ()
29 {
30     int buf[5];

31     buf[-1] = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

32 }
33
34 /*
```

01.w_Defects/underrun_st.c:42

Level Medium

Status Not processed

```
39 {
40     int buf[5];
41     int index = -1;

42     buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

43 }
```

```
44
45 /*
```

Trace

```
vflag == 1
```

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {
202     if (vflag == 1 || vflag ==888)
203     {
204         underrun_st_001();
205     }
```

```
buf[index]
```

01.w_Defects/underrun_st.c:42

```
39 {
40     int buf[5];
41     int index = -1;
42     buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR:Data
Underrun*/
43 }
44
45 /*
```

01.w_Defects/underrun_st.c:55

Level Medium

Status Not processed

```
52 int *p;
53 int ret;
54 p = buf;

55 ret = *(p - 1);/*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

56 }
57
58 /*
```

Trace

```
vflag == 1
```

```
01.w_Defects/underrun_st.c:202
```

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {
```

```
202 if (vflag == 1 || vflag ==888)
```

```
203 {
204     underrun_st_001();
205 }
```

```
*(p - 1)
```

```
01.w_Defects/underrun_st.c:55
```

```
52 int *p;
53 int ret;
54 p = buf;
```

```
55 ret = *(p - 1);/*Tool should detect this line as error*/ /*ERROR:Data
Underrun*/
```

```
56 }
57
58 /*
```

01.w_Defects/underrun_st.c:67

Level Medium**Status** Not processed

```
64 int buf[5];
65 int *p;
66 p = buf;

67 *(p - 1) = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

68 }
69
70 /*
```

Trace

vflag == 1

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {

202 if (vflag == 1 || vflag == 888)

203 {
204     underrun_st_001();
205 }
```

*(p - 1)

01.w_Defects/underrun_st.c:67

```
64 int buf[5];
65 int *p;
66 p = buf;

67 *(p - 1) = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

68 }
```

```
69
70 /*
```

01.w_Defects/underrun_st.c:80

Level Medium

Status Not processed

```
77 int *p;
78 int index = 1;
79 p = buf;

80 *(p - index) = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

81 }
82
83 /*
```

Trace

```
vflag == 1
```

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {

202 if (vflag == 1 || vflag == 888)

203 {
204     underrun_st_001();
205 }
```

*(p - index)

01.w_Defects/underrun_st.c:80

```
77 int *p;
78 int index = 1;
79 p = buf;

80 *(p - index) = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

81 }
82
83 /*
```

01.w_Defects/underrun_st.c:93

Level Medium

Status Not processed

```
90 int i;
91 for (i = 4; i >= -1; i --)
92 {

93         buf[i] = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

94     }
95 }
96
```

Trace

vflag == 1

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {
```

```
202 if (vflag == 1 || vflag ==888)
```

```
203 {  
204     underrun_st_001();  
205 }
```

buf[i]

01.w_Defects/underrun_st.c:93

```
90 int i;  
91 for (i = 4; i >= -1; i --)  
92 {  
  
93     buf[i] = 1; /*Tool should detect this line as error*/ /*ERROR:Data  
Underrun*/  
  
94 }  
95 }  
96
```

01.w_Defects/underrun_st.c:109

Level Medium

Status Not processed

```
106 p = &buf[4];  
107 for (i = 0; i <= 5; i ++)  
108 {  
  
109     *p = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/  
  
110     p --;  
111 }  
112 }
```

Trace

vflag == 1

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {

202     if (vflag == 1 || vflag ==888)

203     {
204         underrun_st_001();
205     }
```

*p

01.w_Defects/underrun_st.c:109

```
106     p = &buf[4];
107     for (i = 0; i <= 5; i++)
108     {

109         *p = 1; /*Tool should detect this line as error*/ /*ERROR:Data
Underrun*/

110         p--;
111     }
112 }
```

01.w_Defects/underrun_st.c:124

Level Medium

Status Not processed

```
121     int i;
122     for (i = 4; i >= -1; i--)
123     {
```

```
124         underrun_st_009_gbl_buf[i] = 1; /*Tool should detect this line as error*/
/*ERROR:Data Underrun*/
```

```
125 }
126 }
127
```

Trace

```
vflag == 1
```

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {
202     if (vflag == 1 || vflag == 888)
203     {
204         underrun_st_001();
205     }
```

```
underrun_st_009_gbl_buf[i]
```

01.w_Defects/underrun_st.c:124

```
121     int i;
122     for (i = 4; i >= -1; i --)
123     {
124         underrun_st_009_gbl_buf[i] = 1; /*Tool should detect this line as
error*/ /*ERROR:Data Underrun*/
125     }
126 }
```

01.w_Defects/underrun_st.c:140

Level Medium

Status Not processed

```
137 p = &underrun_st_010_gbl_buf[4];
138 for (i = 0; i <= 5; i++)
139 {
140     *p = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/
141     p--;
142 }
143 }
```

Trace

```
vflag == 1
```

```
01.w_Defects/underrun_st.c:202
```

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {
```

```
202 if (vflag == 1 || vflag == 888)
```

```
203 {
204     underrun_st_001();
205 }
```

```
*p
```

```
01.w_Defects/underrun_st.c:140
```

```
137 p = &underrun_st_010_gbl_buf[4];
138 for (i = 0; i <= 5; i++)
139 {
```

```
140     *p = 1; /*Tool should detect this line as error*/ /*ERROR:Data
Underrun*/
```

```
141     p--;
142 }
143 }
```

01.w_Defects/underrun_st.c:155

Level Medium**Status** Not processed

```
152 int i=4;
153 while(i >= -1)
154 {
155     underrun_st_011_gbl_buf[i] = 1; /*Tool should detect this line as error*/
/*ERROR:Data Underrun*/
156     i--;
157 }
158 }
```

Trace

vflag == 1

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {
202     if (vflag == 1 || vflag ==888)
203     {
204         underrun_st_001();
205     }
```

underrun_st_011_gbl_buf[i]

01.w_Defects/underrun_st.c:155

```
152 int i=4;
153 while(i >= -1)
154 {
155     underrun_st_011_gbl_buf[i] = 1; /*Tool should detect this line as
error*/ /*ERROR:Data Underrun*/
```

```
156     i--;
157 }
158 }
```

01.w_Defects/underrun_st.c:172

Level Medium

Status Not processed

```
169 int i=4;
170 while(i >= -1)
171 {
172     *p = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/
173     p--;
174     i--;
175 }
```

Trace

vflag == 1

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {
202     if (vflag == 1 || vflag == 888)
203     {
204         underrun_st_001();
205     }
```

```
*p
```

01.w_Defects/underrun_st.c:172

```
169 int i=4;  
170 while(i >= -1)  
171 {  
  
172     *p = 1; /*Tool should detect this line as error*/ /*ERROR:Data Underrun*/  
  
173     p --;  
174     i--;  
175 }
```

01.w_Defects/underrun_st.c:190

Level Medium

Status Not processed

```
187     while(i >= -1)  
188     {  
189         if(var==0)  
  
190             underrun_st_013_gbl_buf[i] = 1; /*Tool should detect this line as error*/  
/*ERROR:Data Underrun*/  
  
191         i--;  
192     }  
193 }
```

Trace

```
vflag == 1
```

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;  
200 void underrun_st_main ()  
201 {
```

```
202 if (vflag == 1 || vflag ==888)
```

```
203 {  
204     underrun_st_001();  
205 }
```

```
underrun_st_013_gbl_buf[i]
```

01.w_Defects/underrun_st.c:190

```
187 while(i >= -1)  
188 {  
189     if(var==0)  
  
190         underrun_st_013_gbl_buf[i] = 1; /*Tool should detect this line as  
error*/ /*ERROR:Data Underrun*/  
  
191     i--;  
192 }  
193 }
```

01.w_Defects/wrong_arguments_func_pointer.c:271

Level Medium

Status Not processed

```
268  
269 void wrong_arguments_func_pointer_011_func_001  
(wrong_arguments_func_pointer_011_s_001* st)  
270 {  
  
271     memset(st, 0, sizeof(*st));  
  
272     st->a = 1;  
273 }  
274
```

Trace

```
vflag == 1
```

01.w_Defects/wrong_arguments_func_pointer.c:610

```
607 extern volatile int vflag;
608 void wrong_arguments_func_pointer_main ()
609 {
610     if (vflag == 1 || vflag == 888)
611     {
612         wrong_arguments_func_pointer_001();
613     }
```

```
memset(st, 0, sizeof(*st))
```

01.w_Defects/wrong_arguments_func_pointer.c:271

```
268
269 void wrong_arguments_func_pointer_011_func_001
(wrong_arguments_func_pointer_011_s_001* st)
270 {
271     memset(st, 0, sizeof(*st));
272     st->a = 1;
273 }
274
```

01.w_Defects/wrong_arguments_func_pointer.c:324

Level Medium

Status Not processed

```
321
322 void wrong_arguments_func_pointer_012_func_001
(wrong_arguments_func_pointer_012_s_001* st)
323 {
```

```
324     memset(st, 0, sizeof(*st));
```

```
325     st->a = 1;  
326 }  
327
```

Trace

vflag == 1

01.w_Defects/wrong_arguments_func_pointer.c:610

```
607 extern volatile int vflag;  
608 void wrong_arguments_func_pointer_main ()  
609 {  
  
610     if (vflag == 1 || vflag ==888)  
  
611     {  
612         wrong_arguments_func_pointer_001();  
613     }
```

memset(st, 0, sizeof(*st))

01.w_Defects/wrong_arguments_func_pointer.c:324

```
321  
322 void wrong_arguments_func_pointer_012_func_001  
(wrong_arguments_func_pointer_012_s_001* st)  
323 {  
  
324     memset(st, 0, sizeof(*st));  
  
325     st->a = 1;  
326 }  
327
```

02.wo_Defects/func_pointer.c:414

Level Medium

Status Not processed

```
411 {  
412   for (j = 0; j < i; j++)  
413   {  
  
414       str_rev[j] = str1[j];  
  
415   }  
416   str_rev[j] = '\0';  
417 }
```

Trace

```
vflag == 1
```

```
02.wo_Defects/func_pointer.c:644
```

```
641 extern volatile int vflag;  
642 void func_pointer_main ()  
643 {  
  
644   if (vflag == 1 || vflag == 888)  
  
645   {  
646       func_pointer_001();  
647   }
```

```
str1[j]
```

```
02.wo_Defects/func_pointer.c:414
```

```
411 {  
412   for (j = 0; j < i; j++)  
413   {  
  
414       str_rev[j] = str1[j];  
  
415   }  
416   str_rev[j] = '\0';  
417 }
```

02.wo_Defects/st_underrun.c:52

Level Medium**Status** Not processed

```
49 {  
50  
51     int len = strlen(s.buf) - 1;  
  
52     for (;s.buf[len] != 'Z';len--) /*Tool should not detect this line as error*/ /* No Stack  
Under RUN error */  
  
53     {  
54         if ( len < 0 )  
55             break;
```

Trace

vflag == 1

02.wo_Defects/st_underrun.c:273

```
270 extern volatile int vflag;  
271 void st_underrun_main ()  
272 {  
  
273     if (vflag == 1 || vflag ==888)  
  
274     {  
275         st_underrun_001();  
276     }
```

s.buf[len]

02.wo_Defects/st_underrun.c:52

```
49 {  
50  
51     int len = strlen(s.buf) - 1;  
  
52     for (;s.buf[len] != 'Z';len--) /*Tool should not detect this line as error*/ /* No  
Stack Under RUN error */
```

```
53  {
54      if ( len < 0 )
55          break;
```

02.wo_Defects/st_underrun.c:236

Level Medium

Status Not processed

```
233 {
234     int len = strlen(s->buf) - 1;
235     char c = 0;

236     for (;s->buf[len] != 'Z';len--)

237     {
238         c = s->buf[len]; /*Tool should not detect this line as error*/ /* No Stack Under
RUN error */
239         if ( len < 0 )
```

Trace

vflag == 1

02.wo_Defects/st_underrun.c:273

```
270 extern volatile int vflag;
271 void st_underrun_main ()
272 {

273     if (vflag == 1 || vflag == 888)

274     {
275         st_underrun_001();
276     }
```

s->buf[len]

02.wo_Defects/st_underrun.c:236

```
233 {  
234     int len = strlen(s->buf) - 1;  
235     char c = 0;  
  
236     for (;s->buf[len] != 'Z';len--)  
  
237     {  
238         c = s->buf[len]; /*Tool should not detect this line as error*/ /* No Stack  
Under RUN error */  
239         if ( len < 0 )
```

02.wo_Defects/st_underrun.c:238

Level Medium

Status Not processed

```
235 char c = 0;  
236 for (;s->buf[len] != 'Z';len--)  
237 {  
  
238     c = s->buf[len]; /*Tool should not detect this line as error*/ /* No Stack Under  
RUN error */  
  
239     if ( len < 0 )  
240         break;  
241 }
```

Trace

vflag == 1

02.wo_Defects/st_underrun.c:273

```
270 extern volatile int vflag;  
271 void st_underrun_main ()  
272 {
```

```
273 if (vflag == 1 || vflag ==888)
```

```
274 {  
275     st_underrun_001();  
276 }
```

s->buf[len]

02.wo_Defects/st_underrun.c:238

```
235 char c = 0;  
236 for (;s->buf[len] != 'Z';len--)  
237 {
```

```
238     c = s->buf[len]; /*Tool should not detect this line as error*/ /* No Stack  
Under RUN error */
```

```
239 if ( len < 0 )  
240 break;  
241 }
```

02.wo_Defects/wrong_arguments_func_pointer.c:409

Level Medium

Status Not processed

```
406 {  
407     a[i] = i;  
408 }
```

```
409 return a[i];
```

```
410 }  
411  
412 void wrong_arguments_func_pointer_014 ()
```

Trace

```
vflag == 1
```

02.wo_Defects/wrong_arguments_func_pointer.c:605

```
602 extern volatile int vflag;
603 void wrong_arguments_func_pointer_main ()
604 {
605     if (vflag == 1 || vflag ==888)
606     {
607         wrong_arguments_func_pointer_001 ();
608     }
}
```

```
a[i]
```

02.wo_Defects/wrong_arguments_func_pointer.c:409

```
406 {
407     a[i] = i;
408 }

409 return a[i];

410 }
411
412 void wrong_arguments_func_pointer_014 ()
```

Type cast error (C/C++)

Description

Incorrect type cast: a pointer to the larger type is cast to a smaller type. This may lead to incorrect behavior of the application. Part of the data stored in the variable will be lost.

Example

In the following example, a pointer to the larger type is cast to a smaller type:

```
void *
intToPointer(short s)
{
    return (void *)s;
```

```
}
```

Recommendations

- Use the correct type cast: do not cast a variable to a smaller type.

Links

1. Casting a large number type to a smaller type — stackoverflow.com

Vulnerability Entries

01.w_Defects/littlemem_st.c:35

Level Medium

Status Not processed

```
32         buf[i] = 1;
33     }
34
35     p = (littlemem_st_001_s_001 *)buf;
36     ret = p->c; /*Tool should detect this line as error*/ /*ERROR:Little Memory or
Overflow*/
37     printf("%d \n",p->c);
38 }
```

Trace

```
(littlemem_st_001_s_001 *)buf
(littlemem_st_001_s_001 *)buf
```

01.w_Defects/littlemem_st.c:35

```
32         buf[i] = 1;
33     }
34
```

```
35     p = (littlemem_st_001_s_001 *)buf;
```

```
36     ret = p->c; /*Tool should detect this line as error*/ /*ERROR:Little Memory or
Overflow*/
37     printf("%d \n",p->c);
38 }
```

```
(littlemem_st_001_s_001 *)buf
(littlemem_st_001_s_001 *)buf
```

01.w_Defects/littlemem_st.c:35

```
32             buf[i] = 1;
33 }
34
```

```
35     p = (littlemem_st_001_s_001 *)buf;
```

```
36     ret = p->c; /*Tool should detect this line as error*/ /*ERROR:Little Memory or
Overflow*/
37     printf("%d \n",p->c);
38 }
```

01.w_Defects/littlemem_st.c:54

Level Medium

Status Not processed

```
51 {
52     char buf[10];
53     littlemem_st_002_s_001 *p;
```

```
54     p = (littlemem_st_002_s_001 *)buf;
```

```
55     p->c = 1; /*Tool should detect this line as error*/ /*ERROR:Little Memory or
Overflow*/
56 }
57
```

Trace

```
(littlemem_st_002_s_001 *)buf  
(littlemem_st_002_s_001 *)buf
```

01.w_Defects/littlemem_st.c:54

```
51 {  
52     char buf[10];  
53     littlemem_st_002_s_001 *p;  
  
54     p = (littlemem_st_002_s_001 *)buf;  
  
55     p->c = 1; /*Tool should detect this line as error*/ /*ERROR:Little Memory or  
Overflow*/  
56 }  
57
```

```
(littlemem_st_002_s_001 *)buf  
(littlemem_st_002_s_001 *)buf
```

01.w_Defects/littlemem_st.c:54

```
51 {  
52     char buf[10];  
53     littlemem_st_002_s_001 *p;  
  
54     p = (littlemem_st_002_s_001 *)buf;  
  
55     p->c = 1; /*Tool should detect this line as error*/ /*ERROR:Little Memory or  
Overflow*/  
56 }  
57
```

01.w_Defects/littlemem_st.c:91

Level Medium

Status Not processed

```
88 {  
89     char buf[10];  
90
```

```
91 littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;  
  
92 littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should detect this line as error*/  
/*ERROR:Little Memory or Overflow*/  
93 }  
94
```

Trace

```
(littlemem_st_004_s_001 *)buf  
(littlemem_st_004_s_001 *)buf
```

01.w_Defects/littlemem_st.c:91

```
88 {  
89     char buf[10];  
90  
  
91 littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;  
  
92 littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should detect this line as  
error*/ /*ERROR:Little Memory or Overflow*/  
93 }  
94
```

```
(littlemem_st_004_s_001 *)buf  
(littlemem_st_004_s_001 *)buf
```

01.w_Defects/littlemem_st.c:91

```
88 {  
89     char buf[10];  
90  
  
91 littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;  
  
92 littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should detect this line as  
error*/ /*ERROR:Little Memory or Overflow*/  
93 }  
94
```

01.w_Defects/littlemem_st.c:111

Level Medium**Status** Not processed

```
108
109 void littlemem_st_005_func_001 ()
110 {

111 littlemem_st_005_s_001_gbl_str = (littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf;

112 }
113
114 void littlemem_st_005 ()
```

Trace

```
(littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf
(littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf
```

01.w_Defects/littlemem_st.c:111

```
108
109 void littlemem_st_005_func_001 ()
110 {

111 littlemem_st_005_s_001_gbl_str = (littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf;

112 }
113
114 void littlemem_st_005 ()
```

```
(littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf
(littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf
```

01.w_Defects/littlemem_st.c:111

```
108
109 void littlemem_st_005_func_001 ()
110 {

111 littlemem_st_005_s_001_gbl_str = (littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf;

112 }
113
114 void littlemem_st_005 ()
```

01.w_Defects/littlemem_st.c:137

Level Medium

Status Not processed

```
134
135 void littlemem_st_006_func_001 ()
136 {

137 littlemem_st_006_s_001_gbl_str = (littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf;

138 }
139
140 void littlemem_st_006_func_002 (int flag)
```

Trace

```
(littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf
(littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf
```

01.w_Defects/littlemem_st.c:137

```
134
135 void littlemem_st_006_func_001 ()
136 {

137 littlemem_st_006_s_001_gbl_str = (littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf;

138 }
139
140 void littlemem_st_006_func_002 (int flag)
```

```
(littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf
(littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf
```

01.w_Defects/littlemem_st.c:137

```
134
135 void littlemem_st_006_func_001 ()
136 {

137 littlemem_st_006_s_001_gbl_str = (littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf;

138 }
139
140 void littlemem_st_006_func_002 (int flag)
```

01.w_Defects/littlemem_st.c:171

Level Medium

Status Not processed

```
169 void littlemem_st_007_func_001 ()  
170 {  
  
171 littlemem_st_007_s_001_gbl_str = (littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf;  
  
172 }  
173  
174 void littlemem_st_007_func_002 (int flag)
```

Trace

```
(littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf  
(littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf
```

01.w_Defects/littlemem_st.c:171

```
168  
169 void littlemem_st_007_func_001 ()  
170 {  
  
171 littlemem_st_007_s_001_gbl_str = (littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf;  
  
172 }  
173  
174 void littlemem_st_007_func_002 (int flag)
```

```
(littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf  
(littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf
```

01.w_Defects/littlemem_st.c:171

```
168  
169 void littlemem_st_007_func_001 ()  
170 {  
  
171 littlemem_st_007_s_001_gbl_str = (littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf;
```

```
172 }
173
174 void littlemem_st_007_func_002 (int flag)
```

01.w_Defects/littlemem_st.c:214

Level Medium

Status Not processed

```
211
212 void littlemem_st_008_func_001 ()
213 {

214 littlemem_st_008_s_001_gbl_str = (littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf;

215 }
216
217 void littlemem_st_008_func_002 (int flag)
```

Trace

```
(littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf
(littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf
```

01.w_Defects/littlemem_st.c:214

```
211
212 void littlemem_st_008_func_001 ()
213 {

214 littlemem_st_008_s_001_gbl_str = (littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf;

215 }
216
```

```
217 void littlemem_st_008_func_002 (int flag)
```

```
(littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf
(littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf
```

01.w_Defects/littlemem_st.c:214

```
211
212 void littlemem_st_008_func_001 ()
213 {

214 littlemem_st_008_s_001_gbl_str = (littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf;

215 }
216
217 void littlemem_st_008_func_002 (int flag)
```

01.w_Defects/littlemem_st.c:258

Level Medium

Status Not processed

```
255
256 void littlemem_st_009_func_001 ()
257 {

258 littlemem_st_009_s_001_gbl_str = (littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf;

259 }
260
261 void littlemem_st_009_func_002 (int flag)
```

Trace

```
(littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf
(littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf
```

01.w_Defects/littlemem_st.c:258

```
255
256 void littlemem_st_009_func_001 ()
257 {

258 littlemem_st_009_s_001_gbl_str = (littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf;

259 }
260
261 void littlemem_st_009_func_002 (int flag)
```

```
(littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf
(littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf
```

01.w_Defects/littlemem_st.c:258

```
255
256 void littlemem_st_009_func_001 ()
257 {

258 littlemem_st_009_s_001_gbl_str = (littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf;

259 }
260
261 void littlemem_st_009_func_002 (int flag)
```

01.w_Defects/littlemem_st.c:300

Level Medium

Status Not processed

```
298 void littlemem_st_010_func_001 ()  
299 {  
  
300 littlemem_st_010_s_001_gbl_str = (littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf;  
  
301 }  
302  
303 void littlemem_st_010_func_002 (int flag)
```

Trace

```
(littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf  
(littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf
```

01.w_Defects/littlemem_st.c:300

```
297  
298 void littlemem_st_010_func_001 ()  
299 {  
  
300 littlemem_st_010_s_001_gbl_str = (littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf;  
  
301 }  
302  
303 void littlemem_st_010_func_002 (int flag)
```

```
(littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf  
(littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf
```

01.w_Defects/littlemem_st.c:300

```
297  
298 void littlemem_st_010_func_001 ()  
299 {  
  
300 littlemem_st_010_s_001_gbl_str = (littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf;
```

```
301 }
302
303 void littlemem_st_010_func_002 (int flag)
```

01.w_Defects/littlemem_st.c:332

Level Medium

Status Not processed

```
329
330 void littlemem_st_011_func_001 ()
331 {

332 littlemem_st_011_s_001_gbl_str = (littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf;

333 }
334
335 void littlemem_st_011_func_002 (int flag)
```

Trace

```
(littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf
(littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf
```

01.w_Defects/littlemem_st.c:332

```
329
330 void littlemem_st_011_func_001 ()
331 {

332 littlemem_st_011_s_001_gbl_str = (littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf;

333 }
334
```

335 void littlemem_st_011_func_002 (int flag)

```
(littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf
(littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf
```

01.w_Defects/littlemem_st.c:332

329

330 void littlemem_st_011_func_001 ()

331 {

332 littlemem_st_011_s_001_gbl_str = (littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf;

333 }

334

335 void littlemem_st_011_func_002 (int flag)

02.wo_Defects/littlemem_st.c:36

Level Medium

Status Not processed

```
33     buf[i] = 1;
34 }
35
```

36 p = (littlemem_st_001_s_001 *)buf;

37 ret = p->c; /*Tool should not detect this line as error*/ /*No ERROR:Little Memory or
Overflow*/
38 printf("%d \n",p->c);
39 }

Trace

```
(littlemem_st_001_s_001 *)buf  
(littlemem_st_001_s_001 *)buf
```

02.wo_Defects/littlemem_st.c:36

```
33         buf[i] = 1;  
34     }  
35  
  
36     p = (littlemem_st_001_s_001 *)buf;  
  
37     ret = p->c; /*Tool should not detect this line as error*/ /*No ERROR:Little  
Memory or Overflow*/  
38     printf("%d \n",p->c);  
39 }
```

```
(littlemem_st_001_s_001 *)buf  
(littlemem_st_001_s_001 *)buf
```

02.wo_Defects/littlemem_st.c:36

```
33         buf[i] = 1;  
34     }  
35  
  
36     p = (littlemem_st_001_s_001 *)buf;  
  
37     ret = p->c; /*Tool should not detect this line as error*/ /*No ERROR:Little  
Memory or Overflow*/  
38     printf("%d \n",p->c);  
39 }
```

02.wo_Defects/littlemem_st.c:55

Level Medium

Status Not processed

```
52 {  
53     char buf[12];  
54     littlemem_st_002_s_001 *p;
```

```
55 p = (littlemem_st_002_s_001 *)buf;  
  
56 p->c = 1; /*Tool should not detect this line as error*/ /*No ERROR:Little Memory or  
Overflow*/  
57 }  
58
```

Trace

```
(littlemem_st_002_s_001 *)buf  
(littlemem_st_002_s_001 *)buf
```

02.wo_Defects/littlemem_st.c:55

```
52 {  
53     char buf[12];  
54     littlemem_st_002_s_001 *p;  
  
55     p = (littlemem_st_002_s_001 *)buf;  
  
56     p->c = 1; /*Tool should not detect this line as error*/ /*No ERROR:Little  
Memory or Overflow*/  
57 }  
58
```

```
(littlemem_st_002_s_001 *)buf  
(littlemem_st_002_s_001 *)buf
```

02.wo_Defects/littlemem_st.c:55

```
52 {  
53     char buf[12];  
54     littlemem_st_002_s_001 *p;  
  
55     p = (littlemem_st_002_s_001 *)buf;  
  
56     p->c = 1; /*Tool should not detect this line as error*/ /*No ERROR:Little  
Memory or Overflow*/  
57 }  
58
```

02.wo_Defects/littlemem_st.c:92

Level Medium**Status** Not processed

```
89 {  
90   char buf[12];  
91  
  
92   littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;  
  
93   littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should not detect this line as error*/  
/*No ERROR:Little Memory or Overflow*/  
94 }  
95
```

Trace

```
(littlemem_st_004_s_001 *)buf  
(littlemem_st_004_s_001 *)buf
```

02.wo_Defects/littlemem_st.c:92

```
89 {  
90   char buf[12];  
91  
  
92   littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;  
  
93   littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should not detect this line as  
error*/ /*No ERROR:Little Memory or Overflow*/  
94 }  
95
```

```
(littlemem_st_004_s_001 *)buf  
(littlemem_st_004_s_001 *)buf
```

02.wo_Defects/littlemem_st.c:92

```
89 {  
90   char buf[12];  
91
```

```
92 littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;  
  
93 littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should not detect this line as  
error*/ /*No ERROR:Little Memory or Overflow*/  
94 }  
95
```

02.wo_Defects/littlemem_st.c:112

Level Medium

Status Not processed

```
109  
110 void littlemem_st_005_func_001 ()  
111 {  
  
112 littlemem_st_005_s_001_gbl_str = (littlemem_st_005_s_001 *)  
littlemem_st_005_gbl_buf;  
  
113 }  
114  
115 void littlemem_st_005 ()
```

Trace

```
(littlemem_st_005_s_001 *)  
littlemem_st_005_gbl_buf  
(littlemem_st_005_s_001 *)  
littlemem_st_005_gbl_buf
```

02.wo_Defects/littlemem_st.c:112

```
109  
110 void littlemem_st_005_func_001 ()  
111 {  
  
112 littlemem_st_005_s_001_gbl_str = (littlemem_st_005_s_001 *)  
littlemem_st_005_gbl_buf;  
  
113 }
```

```
114
115 void littlemem_st_005 ()
```

```
(littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf
(littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf
```

02.wo_Defects/littlemem_st.c:112

```
109
110 void littlemem_st_005_func_001 ()
111 {

112 littlemem_st_005_s_001_gbl_str = (littlemem_st_005_s_001 *)
littlemem_st_005_gbl_buf;

113 }
114
115 void littlemem_st_005 ()
```

02.wo_Defects/littlemem_st.c:138

Level Medium

Status Not processed

```
135
136 void littlemem_st_006_func_001 ()
137 {

138 littlemem_st_006_s_001_gbl_str = (littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf;

139 }
140
141 void littlemem_st_006_func_002 (int flag)
```

Trace

```
(littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf
(littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf
```

02.wo_Defects/littlemem_st.c:138

```
135
136 void littlemem_st_006_func_001 ()
137 {

138 littlemem_st_006_s_001_gbl_str = (littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf;

139 }
140
141 void littlemem_st_006_func_002 (int flag)
```

```
(littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf
(littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf
```

02.wo_Defects/littlemem_st.c:138

```
135
136 void littlemem_st_006_func_001 ()
137 {

138 littlemem_st_006_s_001_gbl_str = (littlemem_st_006_s_001 *)
littlemem_st_006_gbl_buf;

139 }
140
141 void littlemem_st_006_func_002 (int flag)
```

02.wo_Defects/littlemem_st.c:172

Level Medium

Status Not processed

```
170 void littlemem_st_007_func_001 ()  
171 {  
  
172 littlemem_st_007_s_001_gbl_str = (littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf;  
  
173 }  
174  
175 void littlemem_st_007_func_002 (int flag)
```

Trace

```
(littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf  
(littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf
```

02.wo_Defects/littlemem_st.c:172

```
169  
170 void littlemem_st_007_func_001 ()  
171 {  
  
172 littlemem_st_007_s_001_gbl_str = (littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf;  
  
173 }  
174  
175 void littlemem_st_007_func_002 (int flag)
```

```
(littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf  
(littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf
```

02.wo_Defects/littlemem_st.c:172

```
169  
170 void littlemem_st_007_func_001 ()  
171 {  
  
172 littlemem_st_007_s_001_gbl_str = (littlemem_st_007_s_001 *)  
littlemem_st_007_gbl_buf;
```

```
173 }  
174  
175 void littlemem_st_007_func_002 (int flag)
```

02.wo_Defects/littlemem_st.c:215

Level Medium

Status Not processed

```
212  
213 void littlemem_st_008_func_001 ()  
214 {  
  
215 littlemem_st_008_s_001_gbl_str = (littlemem_st_008_s_001 *)  
littlemem_st_008_gbl_buf;  
  
216 }  
217  
218 void littlemem_st_008_func_002 (int flag)
```

Trace

```
(littlemem_st_008_s_001 *)  
littlemem_st_008_gbl_buf  
(littlemem_st_008_s_001 *)  
littlemem_st_008_gbl_buf
```

02.wo_Defects/littlemem_st.c:215

```
212  
213 void littlemem_st_008_func_001 ()  
214 {
```

```
215 littlemem_st_008_s_001_gbl_str = (littlemem_st_008_s_001 *)  
littlemem_st_008_gbl_buf;
```

```
216 }  
217
```

218 void littlemem_st_008_func_002 (int flag)

```
(littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf
(littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf
```

02.wo_Defects/littlemem_st.c:215

```
212
213 void littlemem_st_008_func_001 ()
214 {

215 littlemem_st_008_s_001_gbl_str = (littlemem_st_008_s_001 *)
littlemem_st_008_gbl_buf;

216 }
217
218 void littlemem_st_008_func_002 (int flag)
```

02.wo_Defects/littlemem_st.c:259

Level Medium

Status Not processed

```
256
257 void littlemem_st_009_func_001 ()
258 {

259 littlemem_st_009_s_001_gbl_str = (littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf;

260 }
261
262 void littlemem_st_009_func_002 (int flag)
```

Trace

```
(littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf
(littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf
```

02.wo_Defects/littlemem_st.c:259

```
256
257 void littlemem_st_009_func_001 ()
258 {

259 littlemem_st_009_s_001_gbl_str = (littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf;

260 }
261
262 void littlemem_st_009_func_002 (int flag)
```

```
(littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf
(littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf
```

02.wo_Defects/littlemem_st.c:259

```
256
257 void littlemem_st_009_func_001 ()
258 {

259 littlemem_st_009_s_001_gbl_str = (littlemem_st_009_s_001 *)
littlemem_st_009_gbl_buf;

260 }
261
262 void littlemem_st_009_func_002 (int flag)
```

02.wo_Defects/littlemem_st.c:301

Level Medium

Status Not processed

```
299 void littlemem_st_010_func_001 ()  
300 {  
  
301 littlemem_st_010_s_001_gbl_str = (littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf;  
  
302 }  
303  
304 void littlemem_st_010_func_002 (int flag)
```

Trace

```
(littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf  
(littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf
```

02.wo_Defects/littlemem_st.c:301

```
298  
299 void littlemem_st_010_func_001 ()  
300 {  
  
301 littlemem_st_010_s_001_gbl_str = (littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf;  
  
302 }  
303  
304 void littlemem_st_010_func_002 (int flag)
```

```
(littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf  
(littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf
```

02.wo_Defects/littlemem_st.c:301

```
298  
299 void littlemem_st_010_func_001 ()  
300 {  
  
301 littlemem_st_010_s_001_gbl_str = (littlemem_st_010_s_001 *)  
littlemem_st_010_gbl_buf;
```

```
302 }
303
304 void littlemem_st_010_func_002 (int flag)
```

02.wo_Defects/littlemem_st.c:333

Level Medium

Status Not processed

```
330
331 void littlemem_st_011_func_001 ()
332 {

333 littlemem_st_011_s_001_gbl_str = (littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf;

334 }
335
336 void littlemem_st_011_func_002 (int flag)
```

Trace

```
(littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf
(littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf
```

02.wo_Defects/littlemem_st.c:333

```
330
331 void littlemem_st_011_func_001 ()
332 {

333 littlemem_st_011_s_001_gbl_str = (littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf;

334 }
335
```

```
336 void littlemem_st_011_func_002 (int flag)
```

```
(littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf
(littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf
```

02.wo_Defects/littlemem_st.c:333

```
330
331 void littlemem_st_011_func_001 ()
332 {

333 littlemem_st_011_s_001_gbl_str = (littlemem_st_011_s_001 *)
littlemem_st_011_gbl_buf;

334 }
335
336 void littlemem_st_011_func_002 (int flag)
```

Undefined result (C/C++)

Description

The result of the operation is undefined. Incorrect application behavior is possible.

Example

Incorrect use example:

```
if ([touches count] == 2 && allTouchesEnded) {
    int i = 0;
    int tapCounts[2]; CGPoint tapLocations[2];
    for (UITouch *touch in touches) {
        tapCounts[i] = [touch tapCount];
        tapLocations[i] = [touch locationInView:self];
        i++;
    }
    if (tapCounts[0] == 1 && tapCounts[1] == 1) { // warning
        tapLocation = midpointBetweenPoints(tapLocations[0], tapLocations[1]);
        [self handleTwoFingerTap];
    }
}
```

Recommendations

- Do not use language constructions whose meaning is defined imprecisely.

Links

1. Garbage value issue — iphonedevsdk.com

Vulnerability Entries

01.w_Defects/bit_shift.c:22

Level Medium

Status Not processed

```
19 {  
20     int a = 1;  
21     int ret;  
  
22     ret = a << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/  
  
23     sink = ret;  
24 }  
25
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;  
246 void bit_shift_main ()  
247 {
```

```
248     if (vflag == 1 || vflag == 888)
```

```
249     {
```

```
250     bit_shift_001();  
251 }
```

ret = a << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/

01.w_Defects/bit_shift.c:22

```
19 {  
20   int a = 1;  
21   int ret;  
  
22   ret = a << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/  
  
23   sink = ret;  
24 }  
25
```

01.w_Defects/bit_shift.c:46

Level Medium

Status Not processed

```
43 {  
44   unsigned int a = 1;  
45   unsigned int ret;  
  
46   ret = a << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/  
  
47   sink = ret;  
48 }  
49
```

Trace

```
vflag == 1
```

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;
246 void bit_shift_main ()
247 {
248     if (vflag == 1 || vflag == 888)
249     {
250         bit_shift_001();
251     }
```

```
ret = a << 32; /*Tool should detect this line as
error*/ /*ERROR:Bit shift error*/
```

01.w_Defects/bit_shift.c:46

```
43 {
44     unsigned int a = 1;
45     unsigned int ret;
46     ret = a << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift
error*/
47     sink = ret;
48 }
49
```

01.w_Defects/bit_shift.c:70

Level Medium

Status Not processed

```
67 {
68     int a = 1;
69     int ret;
```

```
70     ret = a << -1; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/
```

```
71     sink = ret;
72 }
73
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;
246 void bit_shift_main ()
247 {
```

```
248 if (vflag == 1 || vflag ==888)
```

```
249 {
250     bit_shift_001();
251 }
```

-1

01.w_Defects/bit_shift.c:70

```
67 {
68     int a = 1;
69     int ret;
```

```
70     ret = a << -1; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/
```

```
71     sink = ret;
72 }
73
```

01.w_Defects/bit_shift.c:82

Level Medium

Status Not processed

```
79 {  
80     int a = 1;  
81     int ret;  
  
82     ret = a >> 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/  
  
83     sink = ret;  
84 }  
85
```

Trace

```
vflag == 1
```

```
01.w_Defects/bit_shift.c:248
```

```
245 extern volatile int vflag;  
246 void bit_shift_main ()  
247 {
```

```
248     if (vflag == 1 || vflag == 888)
```

```
249     {  
250         bit_shift_001();  
251     }
```

```
ret = a >> 32; /*Tool should detect this line as  
error*/ /*ERROR:Bit shift error*/
```

```
01.w_Defects/bit_shift.c:82
```

```
79 {  
80     int a = 1;  
81     int ret;
```

```
82     ret = a >> 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift  
error*/
```

```
83     sink = ret;  
84 }  
85
```

01.w_Defects/bit_shift.c:94

Level Medium**Status** Not processed

```
91 {  
92     int a = 1;  
93     int ret;  
  
94     ret = a >> -1; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/  
  
95     sink = ret;  
96 }  
97
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;  
246 void bit_shift_main ()  
247 {  
  
248     if (vflag == 1 || vflag == 888)  
  
249     {  
250         bit_shift_001();  
251     }
```

-1

01.w_Defects/bit_shift.c:94

```
91 {  
92     int a = 1;  
93     int ret;  
  
94     ret = a >> -1; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/  
  
95     sink = ret;
```

```
96 }  
97
```

01.w_Defects/bit_shift.c:107

Level Medium

Status Not processed

```
104 int a = 1;  
105 int shift = 32;  
106 int ret;
```

```
107 ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/
```

```
108     sink = ret;  
109 }  
110
```

Trace

```
vflag == 1
```

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;  
246 void bit_shift_main ()  
247 {
```

```
248 if (vflag == 1 || vflag == 888)
```

```
249 {  
250     bit_shift_001();  
251 }
```

ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/

01.w_Defects/bit_shift.c:107

```
104 int a = 1;
105 int shift = 32;
106 int ret;

107 ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift
error*/

108     sink = ret;
109 }
110
```

01.w_Defects/bit_shift.c:134

Level Medium

Status Not processed

```
131 int a = 1;
132 int shift = 6;
133 int ret;
```

134 ret = a << ((5 * shift) + 2); /*Tool should detect this line as error*/ /*ERROR:Bit shift
error*/

```
135     sink = ret;
136 }
137
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;
246 void bit_shift_main ()
247 {
248     if (vflag == 1 || vflag == 888)
249     {
250         bit_shift_001();
251     }
```

((5 * shift) + 2)

01.w_Defects/bit_shift.c:134

```
131 int a = 1;
132 int shift = 6;
133 int ret;

134 ret = a << ((5 * shift) + 2); /*Tool should detect this line as error*/ /*ERROR:
Bit shift error*/

135     sink = ret;
136 }
137
```

01.w_Defects/bit_shift.c:147

Level Medium

Status Not processed

```
144 int a = 1;
145 int shift = 5;
146 int ret;
```

```
147 ret = a << ((shift * shift) + 7); /*Tool should detect this line as error*/ /*ERROR: Bit
shift error*/
```

```
148     sink = ret;
149 }
150
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;
246 void bit_shift_main ()
247 {
```

```
248 if (vflag == 1 || vflag ==888)
```

```
249 {
250     bit_shift_001();
251 }
```

((shift * shift) + 7)

01.w_Defects/bit_shift.c:147

```
144 int a = 1;
145 int shift = 5;
146 int ret;
```

```
147 ret = a << ((shift * shift) + 7);/*Tool should detect this line as error*/
/*ERROR:Bit shift error*/
```

```
148     sink = ret;
149 }
150
```

01.w_Defects/bit_shift.c:164

Level Medium

Status Not processed

```
161 {  
162     int a = 1;  
163     int ret;  
  
164     ret = a << bit_shift_012_func_001();/*Tool should detect this line as error*/  
/*ERROR:Bit shift error*/  
  
165     sink = ret;  
166 }  
167
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;  
246 void bit_shift_main ()  
247 {  
  
248     if (vflag == 1 || vflag ==888)  
  
249     {  
250         bit_shift_001();  
251     }
```

bit_shift_012_func_001()

01.w_Defects/bit_shift.c:164

```
161 {  
162     int a = 1;  
163     int ret;  
  
164     ret = a << bit_shift_012_func_001();/*Tool should detect this line as error*/  
/*ERROR:Bit shift error*/  
  
165     sink = ret;  
166 }  
167
```

01.w_Defects/bit_shift.c:176

Level Medium**Status** Not processed

```
173 {  
174     int a = 1;  
175     int ret;  
  
176     ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/  
  
177     sink = ret;  
178 }  
179
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;  
246 void bit_shift_main ()  
247 {  
  
248     if (vflag == 1 || vflag == 888)  
  
249     {  
250         bit_shift_001();  
251     }
```

```
ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/
```

01.w_Defects/bit_shift.c:176

```
173 {  
174     int a = 1;  
175     int ret;  
  
176     ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/
```

```
177     sink = ret;
178 }
179
```

01.w_Defects/bit_shift.c:194

Level Medium

Status Not processed

```
191 int a = 1;
192 int shifts[5] = {8, 40, 16, 32, 24};
193 int ret;

194 ret = a << shifts[3];/*Tool should detect this line as error*/ /*ERROR:Bit shift error*/

195     sink = ret;
196 }
197
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;
246 void bit_shift_main ()
247 {

248 if (vflag == 1 || vflag ==888)

249 {
250     bit_shift_001();
251 }
```

shifts[3]

01.w_Defects/bit_shift.c:194

```
191 int a = 1;
192 int shifts[5] = {8, 40, 16, 32, 24};
193 int ret;

194 ret = a << shifts[3];/*Tool should detect this line as error*/ /*ERROR:Bit shift
error*/

195     sink = ret;
196 }
197
```

01.w_Defects/bit_shift.c:209

Level Medium**Status** Not processed

```
206 int shift1;
207 int ret;
208 shift1 = shift;

209 ret = a << shift1;/*Tool should detect this line as error*/ /*ERROR:Bit shift error*/

210     sink = ret;
211 }
212
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;
246 void bit_shift_main ()
247 {
```

```
248 if (vflag == 1 || vflag ==888)
```

```
249 {  
250     bit_shift_001();  
251 }
```

ret = a << shift1; /*Tool should detect this line
as error*/ /*ERROR:Bit shift error*/

01.w_Defects/bit_shift.c:209

```
206 int shift1;  
207 int ret;  
208 shift1 = shift;
```

```
209 ret = a << shift1; /*Tool should detect this line as error*/ /*ERROR:Bit shift  
error*/
```

```
210     sink = ret;  
211 }  
212
```

01.w_Defects/bit_shift.c:226

Level Medium

Status Not processed

```
223 int ret;  
224 shift1 = shift;  
225 shift2 = shift1;
```

```
226 ret = a << shift2; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/
```

```
227     sink = ret;  
228 }  
229
```

Trace

```
vflag == 1
```

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;
246 void bit_shift_main ()
247 {
248     if (vflag == 1 || vflag == 888)
249     {
250         bit_shift_001();
251     }
```

ret = a << shift2; /*Tool should detect this line
as error*/ /*ERROR:Bit shift error*/

01.w_Defects/bit_shift.c:226

```
223 int ret;
224 shift1 = shift;
225 shift2 = shift1;

226 ret = a << shift2; /*Tool should detect this line as error*/ /*ERROR:Bit shift
error*/

227     sink = ret;
228 }
229
```

01.w_Defects/bit_shift.c:237

Level Medium

Status Not processed

```
234 void bit_shift_017 ()
235 {
236     int ret;
```

237 ret = 1 << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/

```
238     sink = ret;
239 }
240
```

Trace

vflag == 1

01.w_Defects/bit_shift.c:248

```
245 extern volatile int vflag;
246 void bit_shift_main ()
247 {
```

```
248 if (vflag == 1 || vflag ==888)
```

```
249 {
250     bit_shift_001();
251 }
```

ret = 1 << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/

01.w_Defects/bit_shift.c:237

```
234 void bit_shift_017 ()
235 {
236     int ret;
```

```
237     ret = 1 << 32; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/
```

```
238     sink = ret;
239 }
240
```

01.w_Defects/zero_division.c:23

Level Medium

Status Not processed

```
20 {  
21     int dividend = 1000;  
22     int ret;  
  
23     ret = dividend / 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
24 }  
25  
26 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {  
  
262     if (vflag == 1 || vflag == 888)  
  
263     {  
264         zero_division_001();  
265     }
```

dividend / 0

01.w_Defects/zero_division.c:23

```
20 {  
21     int dividend = 1000;  
22     int ret;  
  
23     ret = dividend / 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
24 }  
25  
26 /*
```

01.w_Defects/zero_division.c:34

Level Medium**Status** Not processed

```
31 {  
32     int dividend = 1000;  
33     int ret;  
  
34     dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
35     ret = dividend;  
36 }  
37
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {  
  
262     if (vflag == 1 || vflag == 888)  
  
263     {  
264         zero_division_001();  
265     }
```

dividend /= 0

01.w_Defects/zero_division.c:34

```
31 {  
32     int dividend = 1000;  
33     int ret;  
  
34     dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by  
zero */
```

```
35     ret = dividend;
36 }
37
```

01.w_Defects/zero_division.c:47

Level Medium

Status Not processed

```
44 {
45     int dividend = 1000;
46     int ret;

47     ret = dividend % 0; /*Tool should detect this line as error*/ /* ERROR:division by
zero */

48 }
49
50
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;
260 void zero_division_main ()
261 {

262     if (vflag == 1 || vflag == 888)

263     {
264         zero_division_001();
265     }
```

dividend % 0

01.w_Defects/zero_division.c:47

```
44 {  
45     int dividend = 1000;  
46     int ret;  
  
47     ret = dividend % 0; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */  
  
48 }  
49  
50
```

01.w_Defects/zero_division.c:78

Level Medium

Status Not processed

```
75     int dividend = 1000;  
76     int divisors[5] = {2, 1, 0, 3, 4};  
77     int ret;  
  
78     ret = dividend / divisors[2]; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */  
  
79 }  
80  
81 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

```
262 if (vflag == 1 || vflag ==888)
```

```
263 {  
264     zero_division_001();  
265 }
```

dividend / divisors[2]

01.w_Defects/zero_division.c:78

```
75 int dividend = 1000;  
76 int divisors[5] = {2, 1, 0, 3, 4};  
77 int ret;
```

```
78 ret = dividend / divisors[2];/*Tool should detect this line as error*/ /* ERROR:  
division by zero */
```

```
79 }  
80  
81 /*
```

01.w_Defects/zero_division.c:118

Level Medium

Status Not processed

```
115 int dividend = 1000;  
116 int ret;  
117 zero_division_007_func_001();
```

```
118 ret = dividend / zero_division_007_s_gbl.divisor;/*Tool should detect this line as  
error*/ /* ERROR:division by zero */
```

```
119 }  
120  
121 /*
```

Trace

```
vflag == 1
```

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;
260 void zero_division_main ()
261 {

262     if (vflag == 1 || vflag ==888)

263 {
264     zero_division_001();
265 }
```

```
dividend / zero_division_007_s_gbl.divisor
```

01.w_Defects/zero_division.c:118

```
115 int dividend = 1000;
116 int ret;
117 zero_division_007_func_001();

118 ret = dividend / zero_division_007_s_gbl.divisor; /*Tool should detect this line
as error*/ /* ERROR:division by zero */

119 }
120
121 /*
```

01.w_Defects/zero_division.c:141

Level Medium

Status Not processed

```
138 int dividend = 1000;
139 int divisor = 0;
140 int ret;
```

```
141 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:division by
zero */
```

```
142 }  
143  
144 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

```
262 if (vflag == 1 || vflag ==888)
```

```
263 {  
264     zero_division_001();  
265 }
```

dividend / divisor

01.w_Defects/zero_division.c:141

```
138 int dividend = 1000;  
139 int divisor = 0;  
140 int ret;
```

```
141 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:  
division by zero */
```

```
142 }  
143  
144 /*
```

01.w_Defects/zero_division.c:166

Level Medium

Status Not processed

```
163 int dividend = 1000;
164 int divisor = 2;
165 int ret;

166 ret = dividend / (2 * divisor - 4);/*Tool should detect this line as error*/ /* ERROR:
division by zero */

167 }
168
169 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;
260 void zero_division_main ()
261 {

262 if (vflag == 1 || vflag ==888)

263 {
264     zero_division_001();
265 }
```

dividend / (2 * divisor - 4)

01.w_Defects/zero_division.c:166

```
163 int dividend = 1000;
164 int divisor = 2;
165 int ret;

166 ret = dividend / (2 * divisor - 4);/*Tool should detect this line as error*/ /* ERROR:
division by zero */

167 }
168
169 /*
```

01.w_Defects/zero_division.c:178

Level Medium**Status** Not processed

```
175 int dividend = 1000;  
176 int divisor = 2;  
177 int ret;
```

```
178 ret = dividend / (divisor * divisor - 4);/*Tool should detect this line as error*/ /*  
ERROR:division by zero */
```

```
179  
180 }  
181
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {  
  
262 if (vflag == 1 || vflag ==888)  
  
263 {  
264     zero_division_001();  
265 }
```

dividend / (divisor * divisor - 4)

01.w_Defects/zero_division.c:178

```
175 int dividend = 1000;  
176 int divisor = 2;  
177 int ret;
```

```
178 ret = dividend / (divisor * divisor - 4);/*Tool should detect this line as error*/ /*  
ERROR:division by zero */
```

```
179  
180 }  
181
```

01.w_Defects/zero_division.c:195

Level Medium

Status Not processed

```
192 {  
193   int dividend = 1000;  
194   int ret;
```

195 ret = dividend / zero_division_013_func_001();/*Tool should detect this line as
error*/ /* ERROR:division by zero */

```
196 }  
197  
198 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

262 if (vflag == 1 || vflag == 888)

```
263   {  
264     zero_division_001();  
265   }
```

```
dividend / zero_division_013_func_001()
```

01.w_Defects/zero_division.c:195

```
192 {  
193   int dividend = 1000;  
194   int ret;  
  
195   ret = dividend / zero_division_013_func_001();/*Tool should detect this line  
as error*/ /* ERROR:division by zero */  
  
196 }  
197  
198 /*
```

01.w_Defects/zero_division.c:206

Level Medium

Status Not processed

```
203 {  
204   int dividend = 1000;  
205   int ret;  
  
206   ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:division by  
zero */  
  
207 }  
208  
209 void zero_division_014 ()
```

Trace

```
vflag == 1
```

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

```
262 if (vflag == 1 || vflag ==888)
```

```
263 {  
264     zero_division_001();  
265 }
```

dividend / divisor

01.w_Defects/zero_division.c:206

```
203 {  
204 int dividend = 1000;  
205 int ret;
```

```
206 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:  
division by zero */
```

```
207 }  
208  
209 void zero_division_014 ()
```

01.w_Defects/zero_division.c:225

Level Medium

Status Not processed

```
222 int divisor1;  
223 int ret;  
224 divisor1 = divisor;
```

```
225 ret = dividend / divisor1; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */
```

```
226 }  
227  
228 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;
260 void zero_division_main ()
261 {
262     if (vflag == 1 || vflag == 888)
263     {
264         zero_division_001();
265     }
```

dividend / divisor1

01.w_Defects/zero_division.c:225

```
222 int divisor1;
223 int ret;
224 divisor1 = divisor;
225 ret = dividend / divisor1; /*Tool should detect this line as error*/ /* ERROR:
division by zero */
226 }
227 /*
228 */
```

01.w_Defects/zero_division.c:252

Level Medium

Status Not processed

```
249 zero_division_016_func_002 ();
250 divisor1 = *zero_division_016_gbl_divisor;
251 divisor2 = divisor1;
```

```
252 ret = dividend / divisor2; /*Tool should detect this line as error*/ /* ERROR: division
by zero */
```

```
253 }  
254  
255 /*
```

Trace

vflag == 1

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

```
262 if (vflag == 1 || vflag ==888)
```

```
263 {  
264     zero_division_001();  
265 }
```

dividend / divisor2

01.w_Defects/zero_division.c:252

```
249 zero_division_016_func_002 ();  
250 divisor1 = *zero_division_016_gbl_divisor;  
251 divisor2 = divisor1;
```

```
252 ret = dividend / divisor2; /*Tool should detect this line as error*/ /* ERROR:  
division by zero */
```

```
253 }  
254  
255 /*
```

03.w_Defects_Cpp/improper_error_handling.cpp:22

Level Medium

Status Not processed

```
19 try {  
20     int a=0,b=9,c;  
21     if (a==0)  
  
22     c=b/a;  
  
23 }  
24 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error  
handling*/  
25 {
```

Trace

vflag == 1

```
03.w_Defects_Cpp/improper_error_handling.cpp:108  
  
105 extern volatile int vflag;  
106 void improper_error_handling_main ()  
107 {  
  
108     if (vflag == 1 || vflag ==888)  
  
109     {  
110         improper_error_handling_001();  
111     }
```

b/a

```
03.w_Defects_Cpp/improper_error_handling.cpp:22
```

```
19 try {  
20     int a=0,b=9,c;  
21     if (a==0)  
  
22     c=b/a;  
  
23 }  
24 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error  
handling*/  
25 {
```

Uninitialized variable (C/C++)

Description

An uninitialized variable is used, which causes undefined behavior of the application or points out typos.

In some cases, default value is assigned to a variable which is not initialized properly. This may affect application security depending on the program logic.

Example

In the following example, uninitialized variable z is used:

```
int z;  
int y = z;
```

The correct version:

```
int z = 3;  
int y = z;
```

Recommendations

- When you initialize a variable always assign it some initial value.

Links

1. [CWE-457: Use of Uninitialized Variable](#)

Vulnerability Entries

01.w_Defects/buffer_overrun_dynamic.c:333

Level Medium**Status** Not processed

```
330 int index = 4;
331 if(buf!=NULL)
332 {
333     *(buf+indexes[index]) = 1; /*Tool should detect this line as error*/ /*ERROR:Buffer
overrun*/
334     free(buf);
335 }
336 }
```

Trace

vflag == 1

01.w_Defects/buffer_overrun_dynamic.c:619

```
616 extern volatile int vflag;
617 void dynamic_buffer_overrun_main ()
618 {
619     if (vflag == 1 || vflag ==888)
620     {
621         dynamic_buffer_overrun_001();
622     }
```

indexes[index]

01.w_Defects/buffer_overrun_dynamic.c:333

```
330 int index = 4;
331 if(buf!=NULL)
332 {
333     *(buf+indexes[index]) = 1; /*Tool should detect this line as error*/
/*ERROR:Buffer overrun*/
```

```
334     free(buf);
335 }
336 }
```

01.w_Defects/buffer_underrun_dynamic.c:178

Level Medium

Status Not processed

```
175 int i,j=4;
176 for(i=0;i<5;i++)
177 {
178     *((*pbuff[i-3])+j)=5; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
179 }
180 free(buf1);
181 free(buf2);
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag ==888)
794     {
795         dynamic_buffer_underrun_001();
796     }
```

(*pbuff[i-3])

01.w_Defects/buffer_underrun_dynamic.c:178

```
175 int i,j=4;
176 for(i=0;i<5;i++)
177 {
178     *((*pbuff[i-3])+j)=5; /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
179 }
180 free(buf1);
181 free(buf2);
```

01.w_Defects/buffer_underrun_dynamic.c:648

Level Medium

Status Not processed

```
645 {
646     for(i=-1;i<10;i++)
647     {
648         if(srcbuf[i]==ch) /*Tool should detect this line as error*/ /*ERROR:Buffer Underrun*/
649     {
650         loc=i;
651     }
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
```

```
793 if (vflag == 1 || vflag ==888)

794 {
795     dynamic_buffer_underrun_001();
796 }
```

srcbuf[i]

01.w_Defects/buffer_underrun_dynamic.c:648

```
645 {
646 for(i=-1;i<10;i++)
647 {

648     if(srcbuf[i]==ch) /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/

649     {
650         loc=i;
651     }
```

01.w_Defects/buffer_underrun_dynamic.c:674

Level Medium

Status Not processed

```
671 if (loc1==0)
672 loc1--;
673
```

674 doubleptr[loc1][loc2]='T';

```
675
676 if(loc2==0)
677 loc2--;
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;
791 void dynamic_buffer_underrun_main ()
792 {
793     if (vflag == 1 || vflag == 888)
794     {
795         dynamic_buffer_underrun_001();
796     }
}
```

doubleptr[loc1][loc2]='T'

01.w_Defects/buffer_underrun_dynamic.c:674

```
671 if (loc1 == 0)
672 loc1--;
673
674 doubleptr[loc1][loc2] = 'T';
675
676 if (loc2 == 0)
677 loc2--;
```

01.w_Defects/buffer_underrun_dynamic.c:722

Level Medium

Status Not processed

```
719 for (i=0; i<10; i++)
720 {
721     doubleptr[i-10] = (char*) malloc(10 * sizeof(char)); /*Tool should detect this line as
error*/ /*ERROR:Buffer Underrun*/
722     if (doubleptr[i] != NULL)
```

```
723 {  
724     doubleptr[0][0]='T';  
725     free(doubleptr[i]);
```

Trace

vflag == 1

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {  
  
793     if (vflag == 1 || vflag ==888)  
  
794     {  
795         dynamic_buffer_underrun_001();  
796     }
```

doubleptr[i]

01.w_Defects/buffer_underrun_dynamic.c:722

```
719 for (i=0;i<10;i++)  
720 {  
721     doubleptr[i-10]=(char*) malloc(10*sizeof(char)); /*Tool should detect this line  
as error*/ /*ERROR:Buffer Underrun*/  
  
722     if(doubleptr[i]!=NULL)  
  
723     {  
724         doubleptr[0][0]='T';  
725         free(doubleptr[i]);
```

01.w_Defects/buffer_underrun_dynamic.c:724

Level Medium

Status Not processed

```
721 doubleptr[i-10]=(char*) malloc(10*sizeof(char)); /*Tool should detect this line as  
error*/ /*ERROR:Buffer Underrun*/  
722 if(doubleptr[i]!=NULL)  
723 {  
  
724     doubleptr[0][0]='T';  
  
725     free(doubleptr[i]);  
726 }  
727 }
```

Trace

```
vflag == 1
```

01.w_Defects/buffer_underrun_dynamic.c:793

```
790 extern volatile int vflag;  
791 void dynamic_buffer_underrun_main ()  
792 {  
  
793     if (vflag == 1 || vflag ==888)  
  
794     {  
795         dynamic_buffer_underrun_001();  
796     }
```

```
doubleptr[0][0]='T'
```

01.w_Defects/buffer_underrun_dynamic.c:724

```
721 doubleptr[i-10]=(char*) malloc(10*sizeof(char)); /*Tool should detect this line as  
error*/ /*ERROR:Buffer Underrun*/  
722 if(doubleptr[i]!=NULL)  
723 {  
  
724     doubleptr[0][0]='T';  
  
725     free(doubleptr[i]);  
726 }  
727 }
```

01.w_Defects/func_pointer.c:262

Level Medium**Status** Not processed

```
259 {  
260   for(j=0;j<10;j++)  
261 {  
  
262       doubleptr[i][j] += 1;  
  
263 }  
264 }  
265 doubleptr = (char **)func_pointer_006_func_004();
```

Trace

vflag == 1

01.w_Defects/func_pointer.c:617

```
614 extern volatile int vflag;  
615 void func_pointer_main ()  
616 {  
  
617   if (vflag == 1 || vflag ==888)  
  
618   {  
619       func_pointer_001();  
620   }
```

doubleptr[i][j]

01.w_Defects/func_pointer.c:262

```
259 {  
260   for(j=0;j<10;j++)  
261 {  
  
262       doubleptr[i][j] += 1;  
  
263 }
```

```
264 }
265 doubleptr = (char **)func_pointer_006_func_004();
```

01.w_Defects/func_pointer.c:262

Level Medium

Status Not processed

```
259 {
260   for(j=0;j<10;j++)
261   {
262       doubleptr[i][j] += 1;
263   }
264 }
265 doubleptr = (char **)func_pointer_006_func_004();
```

Trace

vflag == 1

01.w_Defects/func_pointer.c:617

```
614 extern volatile int vflag;
615 void func_pointer_main ()
616 {
617   if (vflag == 1 || vflag ==888)
618   {
619       func_pointer_001();
620   }
```

```
doubleptr[i][j]
```

01.w_Defects/func_pointer.c:262

```
259 {  
260     for(j=0;j<10;j++)  
261     {  
  
262         doubleptr[i][j] += 1;  
  
263     }  
264 }  
265 doubleptr = (char **)func_pointer_006_func_004();
```

01.w_Defects/memory_leak.c:276

Level Medium

Status Not processed

```
273 {  
274     char * buf ;  
275  
  
276     buf = un.u2;  
  
277 }  
278 }  
279
```

Trace

```
vflag == 1
```

01.w_Defects/memory_leak.c:539

```
536 extern volatile int vflag;  
537 void memory_leak_main ()  
538 {  
  
539     if (vflag == 1 || vflag ==888)
```

```
540 {  
541     memory_leak_001();  
542 }
```

un.u2

01.w_Defects/memory_leak.c:276

```
273 {  
274     char * buf;  
275  
  
276     buf = un.u2;  
  
277 }  
278 }  
279
```

01.w_Defects/overrun_st.c:44

Level Medium

Status Not processed

```
41 {  
42     int buf[5] = {1, 2, 3, 4, 5};  
43     int ret;  
  
44     ret = buf[5];/*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
45     sink = buf[idx];  
46 }  
47
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
}
```

buf[5]

01.w_Defects/overrun_st.c:44

```
41 {
42     int buf[5] = {1, 2, 3, 4, 5};
43     int ret;
44     ret = buf[5]; /*Tool should detect this line as error*/ /*ERROR: buffer overrun*/
45     sink = buf[idx];
46 }
47
```

01.w_Defects/overrun_st.c:320

Level Medium

Status Not processed

```
317 int *p;
318 int ret;
319 p = buf;
```

```
320 ret = *(p + 5); /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
321     sink = buf[idx];
322 }
323
```

Trace

vflag == 1

01.w_Defects/overrun_st.c:784

```
781 extern volatile int vflag;
782 void overrun_st_main ()
783 {
784     if (vflag == 1 || vflag == 888)
785     {
786         overrun_st_001();
787     }
```

*(p + 5)

01.w_Defects/overrun_st.c:320

```
317 int *p;
318 int ret;
319 p = buf;

320 ret = *(p + 5); /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */

321     sink = buf[idx];
322 }
323
```

01.w_Defects/ow_memcpy.c:41

Level Medium

Status Not processed

```
38 q = (unsigned char *)dst;
39 for (i = 0; i < size; i++)
40 {

41     *q = *p; /*Tool should detect this line as error*/ /*ERROR:copy of the overlapped
area*/

42     p++;
43     q++;
44 }
```

Trace

vflag ==1

01.w_Defects/ow_memcpy.c:60

```
57 extern volatile int vflag;
58 void ow_memcpy_main ()
59 {

60     if (vflag ==1 || vflag ==888)

61     {
62         ow_memcpy_001();
63     }
```

*p

01.w_Defects/ow_memcpy.c:41

```
38 q = (unsigned char *)dst;
39 for (i = 0; i < size; i++)
40 {

41     *q = *p; /*Tool should detect this line as error*/ /*ERROR:copy of the overlapped
area*/

42     p++;
43     q++;
44 }
```

01.w_Defects/st_underrun.c:227

Level Medium**Status** Not processed

```
224 {  
225   int len = strlen(s->buf) - 1;  
226   char c;  
  
227   for (;s->buf[len] != 'Z';len--)  
  
228   {  
229     c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under RUN error  
*/  
230     /*if (s->buf[len] == '\0')
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;  
261 void st_underrun_main ()  
262 {  
  
263   if (vflag == 1 || vflag ==888)  
  
264   {  
265     st_underrun_001();  
266   }
```

s->buf[len]

01.w_Defects/st_underrun.c:227

```
224 {  
225   int len = strlen(s->buf) - 1;  
226   char c;  
  
227   for (;s->buf[len] != 'Z';len--)
```

```
228  {
229      c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under
RUN error */
230          /*if (s->buf[len] == '\0')
```

01.w_Defects/st_underrun.c:229

Level Medium**Status** Not processed

```
226 char c;
227 for (;s->buf[len] != 'Z';len--)
228 {

229      c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under RUN error
*/
230      /*if (s->buf[len] == '\0')
231      break;*/
232 }
```

Trace

vflag == 1

01.w_Defects/st_underrun.c:263

```
260 extern volatile int vflag;
261 void st_underrun_main ()
262 {

263     if (vflag == 1 || vflag ==888)

264     {
265         st_underrun_001();
266     }
```

```
s->buf[len]
```

01.w_Defects/st_underrun.c:229

```
226 char c;
227 for (;s->buf[len] != 'Z';len--)
228 {

229     c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under
RUN error */

230 /*if (s->buf[len] == '\0')
231     break;*/
232 }
```

01.w_Defects/underrun_st.c:21

Level Medium

Status Not processed

```
18 {
19     int buf[5] = {1, 2, 3, 4, 5};
20     int ret;

21     ret = buf[-1];/*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

22 }
23
24 /*
```

Trace

```
vflag == 1
```

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {
```

```
202 if (vflag == 1 || vflag ==888)
```

```
203 {  
204     underrun_st_001();  
205 }
```

buf[-1]

01.w_Defects/underrun_st.c:21

```
18 {  
19     int buf[5] = {1, 2, 3, 4, 5};  
20     int ret;  
  
21     ret = buf[-1];/*Tool should detect this line as error*/ /*ERROR:Data  
Underrun*/  
  
22 }  
23  
24 /*
```

01.w_Defects/underrun_st.c:55

Level Medium

Status Not processed

```
52     int *p;  
53     int ret;  
54     p = buf;  
  
55     ret = *(p - 1);/*Tool should detect this line as error*/ /*ERROR:Data Underrun*/  
  
56 }  
57  
58 /*
```

Trace

```
vflag == 1
```

01.w_Defects/underrun_st.c:202

```
199 extern volatile int vflag;
200 void underrun_st_main ()
201 {

202     if (vflag == 1 || vflag ==888)

203     {
204         underrun_st_001();
205     }
```

```
*(p - 1)
```

01.w_Defects/underrun_st.c:55

```
52     int *p;
53     int ret;
54     p = buf;

55     ret = *(p - 1);/*Tool should detect this line as error*/ /*ERROR:Data
Underrun*/

56 }
57
58 /*
```

01.w_Defects/uninit_memory_access.c:98

Level Medium

Status Not processed

```
95
96
97     }
```

```
98     k = arr1[1][2][3];/*Tool should detect this line as error*/ /*ERROR:Uninitialized
Memory Access*/
```

```
99 }  
100  
101 /*
```

Trace

vflag == 1

01.w_Defects/uninit_memory_access.c:462

```
459 extern volatile int vflag;  
460 void uninit_memory_access_main ()  
461 {  
  
462 if (vflag == 1 || vflag ==888)  
  
463 {  
464     uninit_memory_access_001();  
465 }
```

arr1[1][2][3]

01.w_Defects/uninit_memory_access.c:98

```
95 }  
96 }  
97 }  
  
98 k = arr1[1][2][3];/*Tool should detect this line as error*/ /*ERROR:  
Uninitialized Memory Access*/  
  
99 }  
100  
101 /*
```

01.w_Defects/uninit_memory_access.c:200

Level Medium

Status Not processed

```
197 {  
198     uninit_memory_access_008_s_001 *s = NULL;  
199     s = uninit_memory_access_008_func_001();  
  
200     s->b = s->a; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Memory  
Access*/  
  
201 }  
202  
203 /*
```

Trace

vflag == 1

01.w_Defects/uninit_memory_access.c:462

```
459 extern volatile int vflag;  
460 void uninit_memory_access_main ()  
461 {  
  
462     if (vflag == 1 || vflag == 888)  
  
463     {  
464         uninit_memory_access_001();  
465     }
```

s->a

01.w_Defects/uninit_memory_access.c:200

```
197 {  
198     uninit_memory_access_008_s_001 *s = NULL;  
199     s = uninit_memory_access_008_func_001();  
  
200     s->b = s->a; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Memory Access*/  
  
201 }  
202  
203 /*
```

01.w_Defects/uninit_memory_access.c:249

Level Medium**Status** Not processed

```
246 {  
247   for(j=0;j<10;j++)  
248   {  
  
249       uninit_memory_access_009_doubleptr_gbl[i][j] += 1; /*Tool should detect  
this line as error*/ /*ERROR:Uninitialized Memory Access*/  
  
250   }  
251   free (uninit_memory_access_009_doubleptr_gbl[i]);  
252   uninit_memory_access_009_doubleptr_gbl[i] = NULL;
```

Trace

vflag == 1

01.w_Defects/uninit_memory_access.c:462

```
459 extern volatile int vflag;  
460 void uninit_memory_access_main ()  
461 {  
  
462   if (vflag == 1 || vflag ==888)  
  
463   {  
464       uninit_memory_access_001();  
465   }
```

uninit_memory_access_009_doubleptr_gbl
[i][j]

01.w_Defects/uninit_memory_access.c:249

```
246 {  
247   for(j=0;j<10;j++)  
248   {  
  
249       uninit_memory_access_009_doubleptr_gbl[i][j] += 1; /*Tool should
```

detect this line as error*/ /*ERROR:Uninitialized Memory Access*/

```
250 }
251 free (uninit_memory_access_009_doubleptr_gbl[i]);
252 uninit_memory_access_009_doubleptr_gbl[i] = NULL;
```

01.w_Defects/uninit_memory_access.c:249

Level Medium

Status Not processed

```
246 {
247 for(j=0;j<10;j++)
248 {
```

249 uninit_memory_access_009_doubleptr_gbl[i][j] += 1; /*Tool should detect
this line as error*/ /*ERROR:Uninitialized Memory Access*/

```
250 }
251 free (uninit_memory_access_009_doubleptr_gbl[i]);
252 uninit_memory_access_009_doubleptr_gbl[i] = NULL;
```

Trace

vflag == 1

01.w_Defects/uninit_memory_access.c:462

```
459 extern volatile int vflag;
460 void uninit_memory_access_main ()
461 {
```

```
462 if (vflag == 1 || vflag ==888)
```

```
463 {
464     uninit_memory_access_001();
465 }
```

**uninit_memory_access_009_doubleptr_gbl
[i][j]**

01.w_Defects/uninit_memory_access.c:249

```
246 {  
247   for(j=0;j<10;j++)  
248   {  
  
249       uninit_memory_access_009_doubleptr_gbl[i][j] += 1; /*Tool should  
detect this line as error*/ /*ERROR:Uninitialized Memory Access*/  
  
250   }  
251   free (uninit_memory_access_009_doubleptr_gbl[i]);  
252   uninit_memory_access_009_doubleptr_gbl[i] = NULL;
```

01.w_Defects/uninit_memory_access.c:298

Level Medium**Status** Not processed

```
295 uninit_memory_access_010_func_001(5);  
296 if(uninit_memory_access_010_s_001_arr_gbl!=NULL)  
297 {  
  
298     ++uninit_memory_access_010_s_001_arr_gbl->data; /*Tool should detect  
this line as error*/ /*ERROR:Uninitialized Memory Access*/  
  
299     free((void *)uninit_memory_access_010_s_001_arr_gbl);  
300 }  
301 }
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_memory_access.c:462

```
459 extern volatile int vflag;
460 void uninit_memory_access_main ()
461 {
462     if (vflag == 1 || vflag == 888)
463     {
464         uninit_memory_access_001();
465     }
```

```
uninit_memory_access_010_s_001_arr_gbl->data
```

01.w_Defects/uninit_memory_access.c:298

```
295     uninit_memory_access_010_func_001(5);
296     if(uninit_memory_access_010_s_001_arr_gbl!=NULL)
297     {
298         ++uninit_memory_access_010_s_001_arr_gbl->data; /*Tool should
detected this line as error*/ /*ERROR:Uninitialized Memory Access*/
299     free((void *)uninit_memory_access_010_s_001_arr_gbl);
300 }
301 }
```

01.w_Defects/uninit_pointer.c:30

Level Medium

Status Not processed

```
27     int a = 5;
28     int *p ;
29     int ret;
```

```
30     ret = *p; /*Tool should detect this line as error*/ /*ERROR:Uninitialized pointer*/
```

```
31 }  
32  
33 /*
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;  
419 void uninit_pointer_main ()  
420 {  
  
421     if (vflag == 1 || vflag ==888)  
  
422     {  
423         uninit_pointer_001();  
424     }
```

*p

01.w_Defects/uninit_pointer.c:30

```
27     int a = 5;  
28     int *p ;  
29     int ret;  
  
30     ret = *p; /*Tool should detect this line as error*/ /*ERROR:Uninitialized  
pointer*/  
  
31 }  
32  
33 /*
```

01.w_Defects/uninit_pointer.c:41

Level Medium

Status Not processed

```
38 {  
39     int a;  
40     int *p ;  
  
41     *p = 1; /*Tool should detect this line as error*/ /*ERROR:Uninitialized pointer*/  
  
42 }  
43  
44 /*
```

Trace

```
vflag == 1
```

```
01.w_Defects/uninit_pointer.c:421
```

```
418 extern volatile int vflag;  
419 void uninit_pointer_main ()  
420 {  
  
421     if (vflag == 1 || vflag ==888)  
  
422     {  
423         uninit_pointer_001();  
424     }
```

```
*p = 1
```

```
01.w_Defects/uninit_pointer.c:41
```

```
38 {  
39     int a;  
40     int *p ;  
  
41     *p = 1; /*Tool should detect this line as error*/ /*ERROR:Uninitialized  
pointer*/  
  
42 }  
43  
44 /*
```

01.w_Defects/uninit_pointer.c:55

Level Medium**Status** Not processed

```
52 int a = 0;
53 int ret;
54 pp = &p;

55 ret = **pp; /*Tool should detect this line as error*/ /*ERROR:Uninitialized pointer*/

56 }
57 /*
58 */
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {

421 if (vflag == 1 || vflag ==888)

422 {
423     uninit_pointer_001();
424 }
```

**pp

01.w_Defects/uninit_pointer.c:55

```
52 int a = 0;
53 int ret;
54 pp = &p;

55 ret = **pp; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/
```

```
56 }  
57  
58 /*
```

01.w_Defects/uninit_pointer.c:90

Level Medium

Status Not processed

```
87 pbuf[3] = buf4;  
88 pbuf[4] = buf5;  
89 int ret;  
  
90 ret = pbuf[1][1];  
  
91 }  
92 void uninit_pointer_005 ()  
93 {
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;  
419 void uninit_pointer_main ()  
420 {  
  
421 if (vflag == 1 || vflag == 888)  
  
422 {  
423     uninit_pointer_001();  
424 }
```

pbuf[1][1]

01.w_Defects/uninit_pointer.c:90

```
87    pbuf[3] = buf4;
88    pbuf[4] = buf5;
89    int ret;

90    ret = pbuf[1][1];

91 }
92 void uninit_pointer_005 ()
93 {
```

01.w_Defects/uninit_pointer.c:104

Level Medium

Status Not processed

```
101 */
102 void uninit_pointer_006_func_001 (int **pp)
103 {

104    **pp = 1; /*Tool should detect this line as error*/ /*ERROR:Uninitialized pointer*/

105 }
106
107 void uninit_pointer_006 ()
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {

421    if (vflag == 1 || vflag == 888)
```

```
422 {
423     uninit_pointer_001();
424 }
```

```
**pp = 1
```

01.w_Defects/uninit_pointer.c:104

```
101 */
102 void uninit_pointer_006_func_001 (int **pp)
103 {

104     **pp = 1; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/

105 }
106
107 void uninit_pointer_006 ()
```

01.w_Defects/uninit_pointer.c:131

Level Medium

Status Not processed

```
128
129 for(i=0;i<5;i++)
130 {
```

```
131     *((*pbuf[i])+j)='a'; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/
```

```
132 }
133     free(buf1);
134     free(buf3);
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {
421     if (vflag == 1 || vflag == 888)
422     {
423         uninit_pointer_001();
424     }
```

*pbuff[i]

01.w_Defects/uninit_pointer.c:131

```
128
129 for(i=0;i<5;i++)
130 {
131     *((*pbuff[i])+j)='a';/*Tool should detect this line as error*/ /*ERROR:
Uninitialized pointer*/
132 }
133     free(buf1);
134     free(buf3);
```

01.w_Defects/uninit_pointer.c:152

Level Medium

Status Not processed

```
149 void uninit_pointer_008_func_001 (uninit_pointer_008_s_001 *p)
150 {
151     int ret;
152     p->uninit=ret;
```

```
153 }  
154 void uninit_pointer_008 ()  
155 {
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;  
419 void uninit_pointer_main ()  
420 {  
  
421 if (vflag == 1 || vflag ==888)  
  
422 {  
423     uninit_pointer_001();  
424 }
```

p->uninit=ret

01.w_Defects/uninit_pointer.c:152

```
149 void uninit_pointer_008_func_001 (uninit_pointer_008_s_001 *p)  
150 {  
151     int ret;  
  
152     p->uninit=ret;  
  
153 }  
154 void uninit_pointer_008 ()  
155 {
```

01.w_Defects/uninit_pointer.c:200

Level Medium

Status Not processed

```
197 /* cast void pointer to a pointer of the appropriate type */
198 char ** cptr = (char **)vptr;
199 char * buf;

200 buf = (*cptr);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/

201 }
202 void uninit_pointer_010 ()
203 {
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {

421 if (vflag == 1 || vflag ==888)

422 {
423     uninit_pointer_001();
424 }
```

*cptr

01.w_Defects/uninit_pointer.c:200

```
197 /* cast void pointer to a pointer of the appropriate type */
198 char ** cptr = (char **)vptr;
199 char * buf;

200 buf = (*cptr);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/

201 }
202 void uninit_pointer_010 ()
203 {
```

01.w_Defects/uninit_pointer.c:200

Level Medium**Status** Not processed

```
197 /* cast void pointer to a pointer of the appropriate type */
198 char ** cptr = (char **)vptr;
199 char * buf;

200 buf = (*cptr);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/
201 }
202 void uninit_pointer_010 ()
203 {
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {

421 if (vflag == 1 || vflag ==888)

422 {
423     uninit_pointer_001();
424 }
```

(*cptr)

01.w_Defects/uninit_pointer.c:200

```
197 /* cast void pointer to a pointer of the appropriate type */
198 char ** cptr = (char **)vptr;
199 char * buf;

200 buf = (*cptr);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/
```

```
201 }
202 void uninit_pointer_010 ()
203 {
```

01.w_Defects/uninit_pointer.c:231

Level Medium

Status Not processed

```
228 {
229   for(i=0; i<10; i++)
230   {

231     a += ptr[i];/*Tool should detect this line as error*/ /*ERROR:Uninitialized pointer*/

232   }
233   break;
234 }
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {

421   if (vflag == 1 || vflag ==888)

422   {
423     uninit_pointer_001();
424 }
```

```
a += ptr[i];/*Tool should detect this line as  
error*/ /*ERROR:Uninitialized pointer*/
```

01.w_Defects/uninit_pointer.c:231

```
228 {  
229   for(i=0; i<10; i++)  
230   {  
  
231     a += ptr[i];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
pointer*/  
  
232   }  
233   break;  
234 }
```

01.w_Defects/uninit_pointer.c:231

Level Medium

Status Not processed

```
228 {  
229   for(i=0; i<10; i++)  
230   {  
  
231     a += ptr[i];/*Tool should detect this line as error*/ /*ERROR:Uninitialized pointer*/  
  
232   }  
233   break;  
234 }
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;  
419 void uninit_pointer_main ()  
420 {
```

```
421 if (vflag == 1 || vflag ==888)
```

```
422 {  
423     uninit_pointer_001();  
424 }
```

ptr[i]

01.w_Defects/uninit_pointer.c:231

```
228 {  
229     for(i=0; i<10; i++)  
230 {
```

```
231     a += ptr[i];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
pointer*/
```

```
232 }  
233 break;  
234 }
```

01.w_Defects/uninit_pointer.c:254

Level Medium

Status Not processed

```
251 }  
252 for(i=0; i<10; i++)  
253 {
```

```
254         arr[i] = ++fptr[i];/*Tool should detect this line as error*/ /*ERROR:  
Uninitialized pointer*/
```

```
255 }  
256 }  
257
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {
421     if (vflag == 1 || vflag ==888)
422     {
423         uninit_pointer_001();
424     }
```

```
++fptr[i]
```

01.w_Defects/uninit_pointer.c:254

```
251 }
252 for(i=0; i<10; i++)
253 {
254     arr[i] = ++fptr[i];/*Tool should detect this line as error*/ /*ERROR:
Uninitialized pointer*/
255 }
256 }
257
```

01.w_Defects/uninit_pointer.c:254

Level Medium

Status Not processed

```
251 }
252 for(i=0; i<10; i++)
253 {
```

```
254     arr[i] = ++fptr[i];/*Tool should detect this line as error*/ /*ERROR:
Uninitialized pointer*/
```

```
255 }  
256 }  
257
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;  
419 void uninit_pointer_main ()  
420 {  
  
421     if (vflag == 1 || vflag ==888)  
  
422     {  
423         uninit_pointer_001();  
424     }
```

fptr[i]

01.w_Defects/uninit_pointer.c:254

```
251 }  
252 for(i=0; i<10; i++)  
253 {  
  
254     arr[i] = ++fptr[i];/*Tool should detect this line as error*/ /*ERROR:  
Uninitialized pointer*/  
  
255 }  
256 }  
257
```

01.w_Defects/uninit_var.c:23

Level Medium

Status Not processed

```
20 {  
21   int a ;  
22   int ret;  
  
23   ret = a; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/  
  
24 }  
25  
26 /*
```

Trace

```
vflag == 1
```

```
01.w_Defects/uninit_var.c:307
```

```
304 extern volatile int vflag;  
305 void uninit_var_main ()  
306 {
```

```
307   if (vflag == 1 || vflag == 888)
```

```
308   {  
309     uninit_var_001();  
310   }
```

```
ret = a; /*Tool should detect this line as  
error*/ /*ERROR:Uninitialized Variable*/
```

```
01.w_Defects/uninit_var.c:23
```

```
20 {  
21   int a ;  
22   int ret;
```

```
23   ret = a; /*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/
```

```
24 }  
25  
26 /*
```

01.w_Defects/uninit_var.c:34

Level Medium**Status** Not processed

```
31 {  
32     int buf[5];  
33     int ret;  
  
34     ret = buf[3];/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/  
  
35 }  
36  
37 /*
```

Trace

vflag == 1

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;  
305 void uninit_var_main ()  
306 {  
  
307     if (vflag == 1 || vflag ==888)  
  
308     {  
309         uninit_var_001();  
310     }
```

buf[3]

01.w_Defects/uninit_var.c:34

```
31 {  
32     int buf[5];  
33     int ret;  
  
34     ret = buf[3];/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/
```

```
35 }  
36  
37 /*
```

01.w_Defects/uninit_var.c:45

Level Medium

Status Not processed

```
42 {  
43     int buf[5][6];  
44     int ret;  
  
45     ret = buf[1][1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/  
  
46 }  
47  
48 /*
```

Trace

vflag == 1

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;  
305 void uninit_var_main ()  
306 {  
  
307     if (vflag == 1 || vflag ==888)  
  
308     {  
309         uninit_var_001();  
310     }
```

buf[1][1]

01.w_Defects/uninit_var.c:45

```
42 {  
43     int buf[5][6];  
44     int ret;  
  
45     ret = buf[1][1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/  
  
46 }  
47  
48 /*
```

01.w_Defects/uninit_var.c:63

Level Medium**Status** Not processed

```
60         dvar1 = 25.8;  
61     else  
62         ;  
  
63     ret = dvar; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/  
  
64 }  
65  
66 /*
```

Trace

vflag == 1

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;  
305 void uninit_var_main ()  
306 {
```

```
307 if (vflag == 1 || vflag ==888)
```

```
308 {  
309     uninit_var_001();  
310 }
```

ret = dvar; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/

01.w_Defects/uninit_var.c:63

```
60     dvar1 = 25.8;  
61 else  
62 ;
```

63 ret = dvar; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/

```
64 }  
65  
66 /*
```

01.w_Defects/uninit_var.c:75

Level Medium

Status Not processed

```
72 int ret;  
73 if (0)  
74     ret = 1;
```

75 return ret; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/

```
76 }  
77  
78 void uninit_var_005 ()
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;
305 void uninit_var_main ()
306 {
307     if (vflag == 1 || vflag == 888)
308     {
309         uninit_var_001();
310     }
}
```

```
return ret; /*Tool should detect this line as
error*/ /*ERROR:Uninitialized Variable*/
```

01.w_Defects/uninit_var.c:75

```
72     int ret;
73     if (0)
74     ret = 1;
75     return ret; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
Variable*/
76 }
77
78 void uninit_var_005 ()
```

01.w_Defects/uninit_var.c:92

Level Medium

Status Not processed

```
89 {
90     long a;
91     int flag = 0;
```

```
92     (flag == 10)? (a = 1):(flag = a); /*Tool should detect this line as error*/ /*ERROR:
Uninitialized Variable*/
```

```
93 }  
94  
95 /*
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;  
305 void uninit_var_main ()  
306 {  
  
307     if (vflag == 1 || vflag ==888)  
  
308     {  
309         uninit_var_001();  
310     }
```

(flag==10)? (a = 1):(flag =a);/*Tool should detect this line as error*/ /*ERROR:
Uninitialized Variable*/

01.w_Defects/uninit_var.c:92

```
89 {  
90     long a;  
91     int flag =0;  
  
92     (flag==10)? (a = 1):(flag =a);/*Tool should detect this line as error*/  
/*ERROR:Uninitialized Variable*/  
  
93 }  
94  
95 /*
```

01.w_Defects/uninit_var.c:111

Level Medium**Status** Not processed

```
108 int ret;
109 s.a = 1;
110 s.b = 1;

111 ret = s.uninit; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
Variable*/

112 }
113
114 /*
```

Trace

vflag == 1

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;
305 void uninit_var_main ()
306 {

307 if (vflag == 1 || vflag ==888)

308 {
309     uninit_var_001();
310 }
```

s.uninit

01.w_Defects/uninit_var.c:111

```
108 int ret;
109 s.a = 1;
110 s.b = 1;

111 ret = s.uninit; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
Variable*/
```

```
112 }  
113  
114 /*
```

01.w_Defects/uninit_var.c:161

Level Medium

Status Not processed

```
158 int i,j;  
159 for (i=0;i<5;i++)  
160     for (j=0;j<6;j++)  
  
161             buf[i][j] = ret[i][j];/*Tool should detect this line as error*/ /*ERROR:  
Uninitialized Variable*/  
  
162  
163 void uninit_var_010 ()  
164 {
```

Trace

vflag == 1

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;  
305 void uninit_var_main ()  
306 {  
  
307     if (vflag == 1 || vflag ==888)  
  
308     {  
309         uninit_var_001();  
310     }
```

ret[i][j]

01.w_Defects/uninit_var.c:161

```
158 int i,j;
159 for (i=0;i<5;i++)
160     for (j=0;j<6;j++)
161         buf[i][j] = ret[i][j];}/*Tool should detect this line as error*/
/*ERROR:Uninitialized Variable*/

162
163 void uninit_var_010 ()
164 {
```

01.w_Defects/uninit_var.c:177

Level Medium**Status** Not processed

```
174 int uninit_var_011_func_001 (int arr1[],int a)
175 {
176     int ret=0;
177     if(arr1[0] > 0)
178         ret = a+arr1[1];/*Tool should detect this line as error*/
/*ERROR:Uninitialized
Variable*/
179     return ret ;
180 }
```

Trace

vflag == 1

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;
305 void uninit_var_main ()
306 {
```

```
307 if (vflag == 1 || vflag ==888)
```

```
308 {  
309     uninit_var_001();  
310 }
```

arr1[0]

01.w_Defects/uninit_var.c:177

```
174 int uninit_var_011_func_001 (int arr1[],int a)  
175 {  
176     int ret=0;
```

```
177     if(arr1[0] > 0)
```

```
178         ret = a+arr1[1];/*Tool should detect this line as error*/ /*ERROR:  
Uninitialized Variable*/  
179         return ret ;  
180 }
```

01.w_Defects/uninit_var.c:178

Level Medium

Status Not processed

```
175 {  
176     int ret=0;  
177     if(arr1[0] > 0)
```

```
178         ret = a+arr1[1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/
```

```
179         return ret ;  
180 }  
181
```

Trace

vflag == 1

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;
305 void uninit_var_main ()
306 {
307     if (vflag == 1 || vflag ==888)
308     {
309         uninit_var_001();
310     }
```

arr1[1]

01.w_Defects/uninit_var.c:178

```
175 {
176     int ret=0;
177     if(arr1[0] > 0)
178         ret = a+arr1[1];/*Tool should detect this line as error*/ /*ERROR:
Uninitialized Variable*/
179     return ret ;
180 }
181
```

01.w_Defects/uninit_var.c:243

Level Medium

Status Not processed

```
240 int uninit_var_013_func_001 (void )
241 {
242     values val ;/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/
243     return val;
```

```
244 }  
245  
246 void uninit_var_013 ()
```

Trace

vflag == 1

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;  
305 void uninit_var_main ()  
306 {  
  
307 if (vflag == 1 || vflag ==888)  
  
308 {  
309     uninit_var_001();  
310 }
```

return val

01.w_Defects/uninit_var.c:243

```
240 int uninit_var_013_func_001 (void )  
241 {  
242 values val ;/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/  
  
243 return val;  
  
244 }  
245  
246 void uninit_var_013 ()
```

01.w_Defects/uninit_var.c:296

Level Medium

Status Not processed

```
293 {  
294     int a[3],ret;  
295     uninit_var_015_func_001(a);  
  
296     ret = a[1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/  
  
297 };  
298  
299
```

Trace

vflag == 1

01.w_Defects/uninit_var.c:307

```
304 extern volatile int vflag;  
305 void uninit_var_main ()  
306 {  
  
307     if (vflag == 1 || vflag ==888)  
  
308     {  
309         uninit_var_001();  
310     }
```

a[1]

01.w_Defects/uninit_var.c:296

```
293 {  
294     int a[3],ret;  
295     uninit_var_015_func_001(a);  
  
296     ret = a[1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/  
  
297 };  
298  
299
```

01.w_Defects/wrong_arguments_func_pointer.c:337

Level Medium**Status** Not processed

```
334 {  
335     st.arr[i] = i;  
336     st1->arr[i] = st.arr[i]+i;  
  
337     temp += st.arr[i];  
  
338 }  
339 }  
340
```

Trace

i = 0

01.w_Defects/wrong_arguments_func_pointer.c:333

```
330 int temp;  
331 int i=0;  
332 memset(st1, 0, sizeof(*st1));  
  
333 for (i = 0; i < MAX; i++)  
  
334 {  
335     st.arr[i] = i;  
336     st1->arr[i] = st.arr[i]+i;
```

temp += st.arr[i]

01.w_Defects/wrong_arguments_func_pointer.c:337

```
334 {  
335     st.arr[i] = i;  
336     st1->arr[i] = st.arr[i]+i;  
  
337     temp += st.arr[i];  
  
338 }
```

```
339 }  
340
```

01.w_Defects/wrong_arguments_func_pointer.c:337

Level Medium**Status** Not processed

```
334 {  
335     st.arr[i] = i;  
336     st1->arr[i] = st.arr[i]+i;  
  
337     temp += st.arr[i];  
  
338 }  
339 }  
340
```

Trace

```
int temp
```

01.w_Defects/wrong_arguments_func_pointer.c:330

```
327  
328 void wrong_arguments_func_pointer_012_func_002  
(wrong_arguments_func_pointer_012_s_001 st,  
wrong_arguments_func_pointer_012_s_001* st1)  
329 {  
  
330     int temp;  
  
331     int i=0;  
332     memset(st1, 0, sizeof(*st1));  
333     for (i = 0; i < MAX; i++)
```

```
temp += st.arr[i]
```

01.w_Defects/wrong_arguments_func_pointer.c:337

```
334 {  
335     st.arr[i] = i;  
336     st1->arr[i] = st.arr[i]+i;  
  
337     temp += st.arr[i];  
  
338 }  
339 }  
340
```

01.w_Defects/zero_division.c:34

Level Medium

Status Not processed

```
31 {  
32     int dividend = 1000;  
33     int ret;  
  
34     dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
35     ret = dividend;  
36 }  
37
```

Trace

```
vflag == 1
```

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;  
260 void zero_division_main ()  
261 {
```

```
262     if (vflag == 1 || vflag == 888)
```

```
263 {  
264     zero_division_001();  
265 }
```

dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */

01.w_Defects/zero_division.c:34

```
31 {  
32     int dividend = 1000;  
33     int ret;  
  
34     dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
35     ret = dividend;  
36 }  
37
```

01.w_Defects/zero_division.c:35

Level Medium

Status Not processed

```
32     int dividend = 1000;  
33     int ret;  
34     dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
35     ret = dividend;  
  
36 }  
37  
38 /*
```

Trace

```
vflag == 1
```

01.w_Defects/zero_division.c:262

```
259 extern volatile int vflag;
260 void zero_division_main ()
261 {
262     if (vflag == 1 || vflag == 888)
263     {
264         zero_division_001();
265     }
```

```
ret = dividend
```

01.w_Defects/zero_division.c:35

```
32     int dividend = 1000;
33     int ret;
34     dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by
zero */
35     ret = dividend;
36 }
37
38 /*
```

02.wo_Defects/ow_memcpy.c:42

Level Medium

Status Not processed

```
39 q = (unsigned char *)dst;
40 for (i = 0; i < size; i++)
41 {
```

```
42     *q = *p; /*Tool should not detect this line as error*/ /*No ERROR:copy of the
overlapped area*/
```

```
43 p++;
44 q++;
45 }
```

Trace

vflag ==1

02.wo_Defects/ow_memcpy.c:61

```
58 extern volatile int vflag;
59 void ow_memcpy_main ()
60 {

61     if (vflag ==1 || vflag ==888)

62     {
63         ow_memcpy_001();
64     }
```

*p

02.wo_Defects/ow_memcpy.c:42

```
39 q = (unsigned char *)dst;
40 for (i = 0; i < size; i++)
41 {

42     *q = *p; /*Tool should not detect this line as error*/ /*No ERROR:copy of the
overlapped area*/

43     p++;
44     q++;
45 }
```

02.wo_Defects/st_underrun.c:236

Level Medium

Status Not processed

```
233 {  
234     int len = strlen(s->buf) - 1;  
235     char c = 0;  
  
236     for (;s->buf[len] != 'Z';len--)  
  
237     {  
238         c = s->buf[len]; /*Tool should not detect this line as error*/ /* No Stack Under  
RUN error */  
239         if ( len < 0 )
```

Trace

vflag == 1

02.wo_Defects/st_underrun.c:273

```
270 extern volatile int vflag;  
271 void st_underrun_main ()  
272 {  
  
273     if (vflag == 1 || vflag ==888)  
  
274     {  
275         st_underrun_001();  
276     }
```

s->buf[len]

02.wo_Defects/st_underrun.c:236

```
233 {  
234     int len = strlen(s->buf) - 1;  
235     char c = 0;  
  
236     for (;s->buf[len] != 'Z';len--)  
  
237     {  
238         c = s->buf[len]; /*Tool should not detect this line as error*/ /* No Stack Under RUN error */  
239         if ( len < 0 )
```

02.wo_Defects/st_underrun.c:238

Level Medium**Status** Not processed

```
235 char c = 0;  
236 for (;s->buf[len] != 'Z';len--)  
237 {
```

```
238     c = s->buf[len]; /*Tool should not detect this line as error*/ /* No Stack Under  
RUN error */
```

```
239 if ( len < 0 )  
240 break;  
241 }
```

Trace

vflag == 1

02.wo_Defects/st_underrun.c:273

```
270 extern volatile int vflag;  
271 void st_underrun_main ()  
272 {
```

```
273 if (vflag == 1 || vflag ==888)
```

```
274 {  
275     st_underrun_001();  
276 }
```

s->buf[len]

02.wo_Defects/st_underrun.c:238

```
235 char c = 0;  
236 for (;s->buf[len] != 'Z';len--)  
237 {
```

```
238     c = s->buf[len]; /*Tool should not detect this line as error*/ /* No Stack  
Under RUN error */
```

```
239 if ( len < 0 )
240 break;
241 }
```

02.wo_Defects/st_underrun.c:242

Level Medium

Status Not processed

```
239     if ( len < 0 )
240         break;
241 }
```

```
242     sink = c;
```

```
243 }
244
```

```
245 void st_underrun_007_func_002 (st_underrun_007_s_001 s)
```

Trace

```
vflag == 1
```

02.wo_Defects/st_underrun.c:273

```
270 extern volatile int vflag;
271 void st_underrun_main ()
272 {
```

```
273     if (vflag == 1 || vflag == 888)
```

```
274     {
275         st_underrun_001();
276     }
```

```
sink = c
```

02.wo_Defects/st_underrun.c:242

```
239     if ( len < 0 )
240         break;
241 }

242     sink = c;

243 }
244
245 void st_underrun_007_func_002 (st_underrun_007_s_001 s)
```

02.wo_Defects/wrong_arguments_func_pointer.c:409

Level Medium

Status Not processed

```
406 {
407     a[i] = i;
408 }

409 return a[i];

410 }
411
412 void wrong_arguments_func_pointer_014 ()
```

Trace

```
vflag == 1
```

02.wo_Defects/wrong_arguments_func_pointer.c:605

```
602 extern volatile int vflag;
603 void wrong_arguments_func_pointer_main ()
604 {

605     if (vflag == 1 || vflag == 888)
```

```
606 {  
607     wrong_arguments_func_pointer_001 ();  
608 }
```

a[i]

02.wo_Defects/wrong_arguments_func_pointer.c:409

```
406 {  
407     a[i] = i;  
408 }  
  
409 return a[i];  
  
410 }  
411  
412 void wrong_arguments_func_pointer_014 ()
```

Unsafe function: rand (C/C++)

Description

The application uses an insecure pseudo-random number generator (PRNG). The generated sequence of numbers is predictable. Examples of insecure PRNG: rand, drand48, erand48, jrand48, lcong48, lrand48, mrand48, nrand48, rand_r, random.

PRNGs generate number sequences based on the initial value of the seed. There are two types of PRNG: statistical and cryptographic. Statistical PRNGs generate predictable sequences, which are similar to random according to the statistical characteristics. They must not be used for security purposes. The result of the cryptographic PRNG, on the contrary, is impossible to predict if the value of seed is derived from a source with high entropy. The value of the current time has a small entropy and is also insecure as a seed. These functions generate predictable sequences and must not be used for information security purposes.

Example

In the following example, the application uses an insecure PRNG:
rand();

Recommendations

- Use secure PRNGs and sources of entropy (/dev/random).

Links

1. CWE-338: Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
2. OWASP: Insecure randomness
3. OWASP Top 10 2017-A3-Sensitive Data Exposure

Vulnerability Entries

01.w_Defects/bit_shift.c:120

Level Medium

Status Not processed

```
117 int a = 1;  
118 int shift;  
119 int ret;
```

```
120 shift = rand();
```

```
121 ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/  
122     sink = ret;  
123 }
```

Trace

shift = rand()

01.w_Defects/bit_shift.c:120

```
117 int a = 1;  
118 int shift;  
119 int ret;
```

```
120 shift = rand();
```

```
121 ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift
```

```
error*/
122     sink = ret;
123 }
```

shift = rand()

01.w_Defects/bit_shift.c:120

```
117 int a = 1;
118 int shift;
119 int ret;

120 shift = rand();
```

```
121 ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift
error*/
122     sink = ret;
123 }
```

01.w_Defects/buffer_underrun_dynamic.c:249

Level Medium

Status Not processed

```
246 {
247     int *buf=(int*) calloc(5,sizeof(int));
248     int index = 5;

249 index = rand()-2;
```

```
250 if(buf!=NULL)
251 {
252
```

Trace

```
index = rand()-2
```

01.w_Defects/buffer_underrun_dynamic.c:249

```
246 {  
247 int *buf=(int*) calloc(5,sizeof(int));  
248 int index = 5;
```

```
249 index = rand()-2;
```

```
250 if(buf!=NULL)  
251 {  
252
```

```
index = rand()-2
```

01.w_Defects/buffer_underrun_dynamic.c:249

```
246 {  
247 int *buf=(int*) calloc(5,sizeof(int));  
248 int index = 5;
```

```
249 index = rand()-2;
```

```
250 if(buf!=NULL)  
251 {  
252
```

01.w_Defects/conflicting_cond.c:23

Level Medium

Status Not processed

```
20 int b = 0;  
21 int ret;  
22
```

```
23 a = rand();
```

```
24 if ((a == 0) && (a == 1))/*Tool should detect this line as error*/ /*ERROR:contradict
```

```
condition*/  
25 {  
26   b += a;
```

Trace

```
a = rand()
```

01.w_Defects/conflicting_cond.c:23

```
20 int b = 0;  
21 int ret;  
22
```

```
23 a = rand();
```

```
24 if ((a == 0) && (a == 1))/*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/
```

```
25 {  
26   b += a;
```

```
a = rand()
```

01.w_Defects/conflicting_cond.c:23

```
20 int b = 0;  
21 int ret;  
22
```

```
23 a = rand();
```

```
24 if ((a == 0) && (a == 1))/*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/
```

```
25 {  
26   b += a;
```

01.w_Defects/conflicting_cond.c:42

Level Medium

Status Not processed

```
39 int b = 0;
40 int ret;
41
42 a = rand();
43 if ((a < 5) && (10 < a))/*Tool should detect this line as error*/ /*ERROR:contradict
condition*/
44 {
45     b += a;
```

Trace

```
a = rand()
```

01.w_Defects/conflicting_cond.c:42

```
39 int b = 0;
40 int ret;
41
42 a = rand();
43 if ((a < 5) && (10 < a))/*Tool should detect this line as error*/ /*ERROR:
contradict condition*/
44 {
45     b += a;
```

```
a = rand()
```

01.w_Defects/conflicting_cond.c:42

```
39 int b = 0;
40 int ret;
41
42 a = rand();
43 if ((a < 5) && (10 < a))/*Tool should detect this line as error*/ /*ERROR:
contradict condition*/
44 {
45     b += a;
```

01.w_Defects/conflicting_cond.c:61

Level Medium**Status** Not processed

```
58 int b = 0;  
59 int ret;  
60
```

```
61 a = rand();
```

```
62 if (((0 < a) && (a < 2)) && ((8 < a) && (a < 10))) /*Tool should detect this line as error*/  
/*ERROR:contradict condition*/  
63 {  
64     b += a;
```

Trace

```
a = rand()
```

01.w_Defects/conflicting_cond.c:61

```
58 int b = 0;  
59 int ret;  
60
```

```
61 a = rand();
```

```
62 if (((0 < a) && (a < 2)) && ((8 < a) && (a < 10))) /*Tool should detect this line as  
error*/ /*ERROR:contradict condition*/  
63 {  
64     b += a;
```

```
a = rand()
```

01.w_Defects/conflicting_cond.c:61

```
58 int b = 0;  
59 int ret;  
60
```

```
61 a = rand();
```

```
62 if (((0 < a) && (a < 2)) && ((8 < a) && (a < 10))) /*Tool should detect this line as  
error*/ /*ERROR:contradict condition*/  
63 {  
64     b += a;
```

01.w_Defects/conflicting_cond.c:80

Level Medium**Status** Not processed

```
77 int b = 0;  
78 int ret;  
79
```

```
80 a = rand();
```

```
81 if (a < 5)  
82 {  
83     if (10 < a) /*Tool should detect this line as error*/ /*ERROR:contradict condition*/
```

Trace

```
a = rand()
```

01.w_Defects/conflicting_cond.c:80

```
77 int b = 0;  
78 int ret;  
79
```

```
80 a = rand();
```

```
81 if (a < 5)  
82 {  
83     if (10 < a) /*Tool should detect this line as error*/ /*ERROR:contradict  
condition*/
```

```
a = rand()
```

01.w_Defects/conflicting_cond.c:80

```
77 int b = 0;  
78 int ret;  
79  
  
80 a = rand();  
  
81 if (a < 5)  
82 {  
83     if (10 < a) /*Tool should detect this line as error*/ /*ERROR:contradict  
condition*/
```

01.w_Defects/conflicting_cond.c:102

Level Medium

Status Not processed

```
99 int b;  
100 int ret;  
101
```

102 a = rand();

```
103 b = ((a == 0) && (a == 1)) ? 0 : 1; /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
104 ret = b;  
105     sink = ret;
```

Trace

```
a = rand()
```

01.w_Defects/conflicting_cond.c:102

```
99 int b;  
100 int ret;  
101
```

```
102 a = rand();  
  
103 b = ((a == 0) && (a == 1)) ? 0 : 1; /*Tool should detect this line as error*/  
/*ERROR:contradict condition*/  
104 ret = b;  
105     sink = ret;
```

a = rand()

01.w_Defects/conflicting_cond.c:102

```
99 int b;  
100 int ret;  
101
```

102 a = rand();

```
103 b = ((a == 0) && (a == 1)) ? 0 : 1; /*Tool should detect this line as error*/  
/*ERROR:contradict condition*/  
104 ret = b;  
105     sink = ret;
```

01.w_Defects/conflicting_cond.c:136

Level Medium

Status Not processed

```
133 int b = 0;  
134 int ret;  
135
```

136 a = rand();

```
137 while ((a == 0) && (a == 1)) /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
138 {  
139     b += a;
```

Trace

```
a = rand()
```

01.w_Defects/conflicting_cond.c:136

```
133 int b = 0;  
134 int ret;  
135
```

```
136 a = rand();
```

```
137 while ((a == 0) && (a == 1)) /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
138 {  
139   b += a;
```

```
a = rand()
```

01.w_Defects/conflicting_cond.c:136

```
133 int b = 0;  
134 int ret;  
135
```

```
136 a = rand();
```

```
137 while ((a == 0) && (a == 1)) /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
138 {  
139   b += a;
```

01.w_Defects/conflicting_cond.c:156

Level Medium

Status Not processed

```
153 int b = 0;  
154 int ret;  
155
```

```
156 a = rand();
```

```
157 while ((a < 5) && (10 < a)) /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
158 {  
159   b += a;
```

Trace

```
a = rand()
```

01.w_Defects/conflicting_cond.c:156

```
153 int b = 0;  
154 int ret;  
155
```

156 a = rand();

```
157 while ((a < 5) && (10 < a)) /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
158 {  
159   b += a;
```

```
a = rand()
```

01.w_Defects/conflicting_cond.c:156

```
153 int b = 0;  
154 int ret;  
155
```

156 a = rand();

```
157 while ((a < 5) && (10 < a)) /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
158 {  
159   b += a;
```

01.w_Defects/conflicting_cond.c:176

Level Medium

Status Not processed

```
173 int b = 0;  
174 int ret;  
175
```

```
176 a = rand();
```

```
177 while (((0 < a) && (a < 2)) && ((8 < a) && (a < 10))) /*Tool should detect this line as  
error*/ /*ERROR:contradict condition*/  
178 {  
179   b += a;
```

Trace

```
a = rand()
```

```
01.w_Defects/conflicting_cond.c:176
```

```
173 int b = 0;  
174 int ret;  
175
```

```
176 a = rand();
```

```
177 while (((0 < a) && (a < 2)) && ((8 < a) && (a < 10))) /*Tool should detect this  
line as error*/ /*ERROR:contradict condition*/
```

```
178 {  
179   b += a;
```

```
a = rand()
```

```
01.w_Defects/conflicting_cond.c:176
```

```
173 int b = 0;  
174 int ret;  
175
```

```
176 a = rand();
```

```
177 while (((0 < a) && (a < 2)) && ((8 < a) && (a < 10))) /*Tool should detect this  
line as error*/ /*ERROR:contradict condition*/  
178 {
```

```
179 b += a;
```

01.w_Defects/conflicting_cond.c:197

Level Medium

Status Not processed

```
194  
195 do  
196 {
```

```
197 a = rand();
```

```
198 }  
199 while ((a == 0) && (a == 1)); /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
200 ret = a;
```

Trace

```
a = rand()
```

01.w_Defects/conflicting_cond.c:197

```
194  
195 do  
196 {
```

```
197 a = rand();
```

```
198 }  
199 while ((a == 0) && (a == 1)); /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
200 ret = a;
```

```
a = rand()
```

01.w_Defects/conflicting_cond.c:197

```
194  
195 do  
196 {  
  
197 a = rand();  
  
198 }  
199 while ((a == 0) && (a == 1)); /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
200 ret = a;
```

01.w_Defects/data_lost.c:157

Level Medium

Status Not processed

```
154 {  
155 short ret;  
156 int a;
```

157 a = rand();

```
158 ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost  
because of cast*/  
159     sink = ret;  
160 }
```

Trace

```
a = rand()
```

01.w_Defects/data_lost.c:157

```
154 {  
155 short ret;  
156 int a;
```

157

`a = rand();`

```
158 ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision  
lost because of cast*/  
159     sink = ret;  
160 }
```

`a = rand()`

01.w_Defects/data_lost.c:157

```
154 {  
155     short ret;  
156     int a;
```

157

`a = rand();`

```
158 ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision  
lost because of cast*/  
159     sink = ret;  
160 }
```

01.w_Defects/data_overflow.c:204

Level Medium**Status** Not processed

```
201     int max = 0x7fffffff;  
202     int d;  
203     int ret;
```

204

`d = rand();`

```
205     ret = max + d; /*Tool should detect this line as error*/ /*ERROR:Data Overflow*/  
206     sink = ret;  
207 }
```

Trace

```
d = rand()
```

01.w_Defects/data_overflow.c:204

```
201 int max = 0x7fffffff;
202 int d;
203 int ret;

204 d = rand();

205 ret = max + d; /*Tool should detect this line as error*/ /*ERROR:Data
Overflow*/
206     sink = ret;
207 }
```

```
d = rand()
```

01.w_Defects/data_overflow.c:204

```
201 int max = 0x7fffffff;
202 int d;
203 int ret;

204 d = rand();

205 ret = max + d; /*Tool should detect this line as error*/ /*ERROR:Data
Overflow*/
206     sink = ret;
207 }
```

01.w_Defects/double_free.c:81

Level Medium

Status Not processed

```
78 *(ptr+i)='a';
79 }
80
```

```
81 if (rand() % 2==0)
```

```
82 {  
83     free(ptr);  
84 }
```

Trace

```
if (rand() % 2==0)
```

01.w_Defects/double_free.c:81

```
78     *(ptr+i)='a';  
79 }  
80
```

```
81 if (rand() % 2==0)
```

```
82 {  
83     free(ptr);  
84 }
```

```
if (rand() % 2==0)
```

01.w_Defects/double_free.c:81

```
78     *(ptr+i)='a';  
79 }  
80
```

```
81 if (rand() % 2==0)
```

```
82 {  
83     free(ptr);  
84 }
```

01.w_Defects/double_free.c:86

Level Medium

Status Not processed

```
83     free(ptr);
84 }
85

86 if(rand() % 3==0)

87 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
88 }
89
```

Trace

```
if(rand() % 3==0)
```

01.w_Defects/double_free.c:86

```
83     free(ptr);
84 }
85

86 if(rand() % 3==0)
```

```
87 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
88 }
89
```

```
if(rand() % 3==0)
```

01.w_Defects/double_free.c:86

```
83     free(ptr);
84 }
85

86 if(rand() % 3==0)
```

```
87 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
88 }
89
```

01.w_Defects/double_release.c:54

Level Medium**Status** Not processed

```
51 {  
52   while (1)  
53   {  
  
54     if (rand ())  
  
55     {  
56       double_release_001_tsk_001 (NULL);  
57     }  
}
```

Trace

if (rand ())

01.w_Defects/double_release.c:54

```
51 {  
52   while (1)  
53   {  
  
54     if (rand ())  
  
55     {  
56       double_release_001_tsk_001 (NULL);  
57     }  
}
```

if (rand ())

01.w_Defects/double_release.c:54

```
51 {  
52   while (1)  
53   {  
  
54     if (rand ())  
  
55     {  
56       double_release_001_tsk_001 (NULL);  
57     }  
}
```

```
56         double_release_001_tsk_001 (NULL);  
57     }
```

01.w_Defects/double_release.c:105

Level Medium

Status Not processed

```
102 {  
103   while (1)  
104   {  
  
105       if (rand ())  
  
106       {  
107           double_release_002_tsk_001 (NULL);  
108       }  
109   }  
110 }
```

Trace

```
if (rand ())
```

01.w_Defects/double_release.c:105

```
102 {  
103   while (1)  
104   {  
  
105       if (rand ())  
  
106       {  
107           double_release_002_tsk_001 (NULL);  
108       }  
109   }  
110 }
```

```
if (rand ())
```

01.w_Defects/double_release.c:105

```
102 {  
103   while (1)  
104   {  
  
105       if (rand ())  
  
106       {  
107           double_release_002_tsk_001 (NULL);  
108       }  
}
```

01.w_Defects/double_release.c:151

Level Medium

Status Not processed

```
148 {  
149   while (1)  
150   {  
  
151       if (rand ())  
  
152       {  
153           double_release_003_tsk_001 (NULL);  
154       }  
}
```

Trace

```
if (rand ())
```

01.w_Defects/double_release.c:151

```
148 {  
149   while (1)  
150   {  
  
151       if (rand ())
```

```
152     {  
153         double_release_003_tsk_001 (NULL);  
154     }
```

```
if (rand ())
```

01.w_Defects/double_release.c:151

```
148 {  
149 while (1)  
150 {  
  
151     if (rand ())  
  
152     {  
153         double_release_003_tsk_001 (NULL);  
154     }
```

01.w_Defects/double_release.c:196

Level Medium

Status Not processed

```
193 {  
194 while (1)  
195 {  
  
196     if (rand ())  
  
197     {  
198         double_release_004_tsk_001 (NULL);  
199     }
```

Trace

```
if (rand ())
```

01.w_Defects/double_release.c:196

```
193 {  
194   while (1)  
195 {  
  
196       if (rand ())  
  
197       {  
198           double_release_004_tsk_001 (NULL);  
199       }
```

```
if (rand ())
```

01.w_Defects/double_release.c:196

```
193 {  
194   while (1)  
195 {  
  
196       if (rand ())  
  
197       {  
198           double_release_004_tsk_001 (NULL);  
199       }
```

01.w_Defects/double_release.c:245

Level Medium

Status Not processed

```
242 {  
243   while (1)  
244 {  
  
245       if (rand ())  
  
246       {
```

```
247         double_release_005_tsk_001 (NULL);  
248     }
```

Trace

```
if (rand ())
```

01.w_Defects/double_release.c:245

```
242 {  
243 while (1)  
244 {  
  
245     if (rand ())  
  
246     {  
247         double_release_005_tsk_001 (NULL);  
248     }
```

```
if (rand ())
```

01.w_Defects/double_release.c:245

```
242 {  
243 while (1)  
244 {  
  
245     if (rand ())  
  
246     {  
247         double_release_005_tsk_001 (NULL);  
248     }
```

01.w_Defects/double_release.c:281

Level Medium

Status Not processed

```
278 pthread_create (& tid1, NULL, double_release_006_tsk_001, NULL);
```

```
279 pthread_join (tid1, NULL);
280

281 if(rand())

282 pthread_mutex_unlock (double_release_006_glb_mutex);
283 pthread_mutex_unlock (double_release_006_glb_mutex);/*Tool should detect this
line as error*/ /*ERROR:Double UnLock*/
284 pthread_mutex_destroy (double_release_006_glb_mutex);
```

Trace

```
if(rand())
```

01.w_Defects/double_release.c:281

```
278 pthread_create (& tid1, NULL, double_release_006_tsk_001, NULL);
279 pthread_join (tid1, NULL);
280
```

```
281 if(rand())
```

```
282 pthread_mutex_unlock (double_release_006_glb_mutex);
283 pthread_mutex_unlock (double_release_006_glb_mutex);/*Tool should
detect this line as error*/ /*ERROR:Double UnLock*/
284 pthread_mutex_destroy (double_release_006_glb_mutex);
```

```
if(rand())
```

01.w_Defects/double_release.c:281

```
278 pthread_create (& tid1, NULL, double_release_006_tsk_001, NULL);
279 pthread_join (tid1, NULL);
280
```

```
281 if(rand())
```

```
282 pthread_mutex_unlock (double_release_006_glb_mutex);
283 pthread_mutex_unlock (double_release_006_glb_mutex);/*Tool should
detect this line as error*/ /*ERROR:Double UnLock*/
284 pthread_mutex_destroy (double_release_006_glb_mutex);
```

01.w_Defects/double_release.c:292

Level Medium**Status** Not processed

```
289 {  
290   while (1)  
291   {  
  
292       if (rand ())  
  
293       {  
294           double_release_006_tsk_001 (NULL);  
295       }  
}
```

Trace

if (rand ())

01.w_Defects/double_release.c:292

```
289 {  
290   while (1)  
291   {  
  
292       if (rand ())  
  
293       {  
294           double_release_006_tsk_001 (NULL);  
295       }  
}
```

if (rand ())

01.w_Defects/double_release.c:292

```
289 {  
290   while (1)  
291   {  
  
292       if (rand ())  
  
293       {  
}
```

```
294         double_release_006_tsk_001 (NULL);  
295     }
```

01.w_Defects/free_null_pointer.c:392

Level Medium

Status Not processed

```
389  
390 free_null_pointer_011_u_001 * free_null_pointer_011_func_002 ()  
391 {
```

```
392 int flag = rand();
```

```
393 switch (flag)  
394 {  
395     case 1:
```

Trace

```
int flag = rand()
```

01.w_Defects/free_null_pointer.c:392

```
389  
390 free_null_pointer_011_u_001 * free_null_pointer_011_func_002 ()  
391 {
```

```
392 int flag = rand();
```

```
393 switch (flag)  
394 {  
395     case 1:
```

```
int flag = rand()
```

01.w_Defects/free_null_pointer.c:392

```
389  
390 free_null_pointer_011_u_001 * free_null_pointer_011_func_002 ()  
391 {  
  
392 int flag = rand();  
  
393 switch (flag)  
394 {  
395     case 1:
```

01.w_Defects/func_pointer.c:330

Level Medium

Status Not processed

```
327  
328 func_pointer_009_u_001 * func_pointer_009_func_001 (void)  
329 {  
  
330 int flag = rand();  
  
331 flag = 1;  
332 func_pointer_009_u_001 *u;  
333 switch (flag)
```

Trace

```
int flag = rand()
```

01.w_Defects/func_pointer.c:330

```
327  
328 func_pointer_009_u_001 * func_pointer_009_func_001 (void)  
329 {  
  
330 int flag = rand();
```

```
331 flag = 1;
332 func_pointer_009_u_001 *u;
333 switch (flag)
```

```
int flag = rand()
```

01.w_Defects/func_pointer.c:330

```
327
328 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
329 {

330 int flag = rand();

331 flag = 1;
332 func_pointer_009_u_001 *u;
333 switch (flag)
```

01.w_Defects/insign_code.c:22

Level Medium

Status Not processed

```
19 int i;
20 int j;
21

22 i = rand();

23 j = i - 1;
24 i = j + 1; /*Tool should detect this line as error*/ /*ERROR:Useless Assignment */
25 }
```

Trace

```
i = rand()
```

01.w_Defects/insign_code.c:22

```
19 int i;  
20 int j;  
21  
  
22 i = rand();  
  
23 j = i - 1;  
24 i = j + 1; /*Tool should detect this line as error*/ /*ERROR:Useless  
Assignment */  
25 }
```

```
i = rand()
```

01.w_Defects/insign_code.c:22

```
19 int i;  
20 int j;  
21  
  
22 i = rand();  
  
23 j = i - 1;  
24 i = j + 1; /*Tool should detect this line as error*/ /*ERROR:Useless  
Assignment */  
25 }
```

01.w_Defects/memory_allocation_failure.c:150

Level Medium

Status Not processed

```
147 void memory_allocation_failure_005 ()  
148 {  
149 int ret;  
  
150 ret = memory_allocation_failure_005_func_001 (rand());
```

```
151 if(ret >= 0)
152     if(vptr != NULL)
153         free(vptr);
```

Trace

```
ret =
memory_allocation_failure_005_func_001
(rand())
```

01.w_Defects/memory_allocation_failure.c:150

```
147 void memory_allocation_failure_005 ()
148 {
149     int ret;

150     ret = memory_allocation_failure_005_func_001 (rand());

151     if(ret >= 0)
152         if(vptr != NULL)
153             free(vptr);
```

```
ret =
memory_allocation_failure_005_func_001
(rand())
```

01.w_Defects/memory_allocation_failure.c:150

```
147 void memory_allocation_failure_005 ()
148 {
149     int ret;

150     ret = memory_allocation_failure_005_func_001 (rand());

151     if(ret >= 0)
152         if(vptr != NULL)
153             free(vptr);
```

01.w_Defects/memory_leak.c:197

Level Medium

Status Not processed

```
194 void memory_leak_007 ()  
195 {  
196     int ret;  
  
197     ret = memory_leak_007_func_001 (rand());  
  
198     if(ret == 0)  
199         if(vptr!=NULL)  
200             {
```

Trace

```
ret = memory_leak_007_func_001 (rand())
```

01.w_Defects/memory_leak.c:197

```
194 void memory_leak_007 ()  
195 {  
196     int ret;  
  
197     ret = memory_leak_007_func_001 (rand());  
  
198     if(ret == 0)  
199         if(vptr!=NULL)  
200             {
```

```
ret = memory_leak_007_func_001 (rand())
```

01.w_Defects/memory_leak.c:197

```
194 void memory_leak_007 ()  
195 {  
196     int ret;  
  
197     ret = memory_leak_007_func_001 (rand());  
  
198     if(ret == 0)  
199         if(vptr!=NULL)  
200             {
```

01.w_Defects/not_return.c:29

Level Medium**Status** Not processed

```
26 void not_return_001 ()  
27 {  
28     int ret;  
  
29     ret = not_return_001_func_001(rand());  
  
30     sink = ret;  
31 }  
32
```

Trace

```
ret = not_return_001_func_001(rand())
```

01.w_Defects/not_return.c:29

```
26 void not_return_001 ()  
27 {  
28     int ret;  
  
29     ret = not_return_001_func_001(rand());  
  
30     sink = ret;  
31 }  
32
```

```
ret = not_return_001_func_001(rand())
```

01.w_Defects/not_return.c:29

```
26 void not_return_001 ()  
27 {  
28     int ret;  
  
29     ret = not_return_001_func_001(rand());  
  
30     sink = ret;  
31 }  
32
```

01.w_Defects/not_return.c:55

Level Medium

Status Not processed

```
52 void not_return_002 ()  
53 {  
54     int ret;  
  
55     ret = not_return_002_func_001(rand(), rand());  
  
56     sink = ret;  
57 }  
58
```

Trace

```
ret = not_return_002_func_001(rand(),  
rand())
```

01.w_Defects/not_return.c:55

```
52 void not_return_002 ()  
53 {  
54     int ret;
```

```
55     ret = not_return_002_func_001(rand(), rand());  
  
56     sink = ret;  
57 }  
58
```

```
ret = not_return_002_func_001(rand(),  
rand())
```

01.w_Defects/not_return.c:55

```
52 void not_return_002 ()  
53 {  
54     int ret;  
  
55     ret = not_return_002_func_001(rand(), rand());  
  
56     sink = ret;  
57 }  
58
```

01.w_Defects/not_return.c:81

Level Medium

Status Not processed

```
78 void not_return_003 ()  
79 {  
80     int ret;  
  
81     ret = not_return_003_func_001(rand());  
  
82     sink = ret;  
83 }  
84
```

Trace

```
ret = not_return_003_func_001(rand())
```

01.w_Defects/not_return.c:81

```
78 void not_return_003 ()  
79 {  
80     int ret;  
  
81     ret = not_return_003_func_001(rand());  
  
82     sink = ret;  
83 }  
84
```

```
ret = not_return_003_func_001(rand())
```

01.w_Defects/not_return.c:81

```
78 void not_return_003 ()  
79 {  
80     int ret;  
  
81     ret = not_return_003_func_001(rand());  
  
82     sink = ret;  
83 }  
84
```

01.w_Defects/not_return.c:104

Level Medium

Status Not processed

```
101 void not_return_004 ()  
102 {  
103     int ret;  
  
104     ret = not_return_004_func_001(rand());  
  
105     sink = ret;
```

```
106 }  
107
```

Trace

```
ret = not_return_004_func_001(rand())
```

01.w_Defects/not_return.c:104

```
101 void not_return_004 ()  
102 {  
103     int ret;  
  
104     ret = not_return_004_func_001(rand());  
  
105     sink = ret;  
106 }  
107
```

```
ret = not_return_004_func_001(rand())
```

01.w_Defects/not_return.c:104

```
101 void not_return_004 ()  
102 {  
103     int ret;  
  
104     ret = not_return_004_func_001(rand());  
  
105     sink = ret;  
106 }  
107
```

01.w_Defects/null_pointer.c:104

Level Medium

Status Not processed

```
101 void null_pointer_006 ()
```

```
102 {  
103   int *p;  
  
104   p = (int *)(intptr_t)rand();  
  
105   *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/  
106 }  
107
```

Trace

```
p = (int *)(intptr_t)rand()
```

01.w_Defects/null_pointer.c:104

```
101 void null_pointer_006 ()  
102 {  
103   int *p;  
  
104   p = (int *)(intptr_t)rand();
```

```
105   *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
derefence*/  
106 }  
107
```

```
p = (int *)(intptr_t)rand()
```

01.w_Defects/null_pointer.c:104

```
101 void null_pointer_006 ()  
102 {  
103   int *p;  
  
104   p = (int *)(intptr_t)rand();
```

```
105   *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer  
derefence*/  
106 }  
107
```

01.w_Defects/overrun_st.c:181

Level Medium**Status** Not processed

```
178 {  
179 int buf[5];  
180 int index;  
  
181 index = rand();  
  
182 buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
183     sink = buf[idx];  
184 }
```

Trace

index = rand()

01.w_Defects/overrun_st.c:181

```
178 {  
179 int buf[5];  
180 int index;  
  
181 index = rand();  
  
182 buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
183     sink = buf[idx];  
184 }
```

index = rand()

01.w_Defects/overrun_st.c:181

```
178 {  
179 int buf[5];  
180 int index;  
  
181 index = rand();
```

```
182 buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */
183     sink = buf[idx];
184 }
```

01.w_Defects/overrun_st.c:442

Level Medium**Status** Not processed

```
439 int *p;
440 int index;
441 p = buf;
```

442 index = rand();

```
443 *(p + index) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
444     sink = buf[idx];
445 }
```

Trace

index = rand()

01.w_Defects/overrun_st.c:442

```
439 int *p;
440 int index;
441 p = buf;
```

442 index = rand();

```
443 *(p + index) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */
444     sink = buf[idx];
445 }
```

```
index = rand()
```

01.w_Defects/overrun_st.c:442

```
439 int *p;
440 int index;
441 p = buf;

442 index = rand();

443 *(p + index) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer
overrun */
444     sink = buf[idx];
445 }
```

01.w_Defects/redundant_cond.c:25

Level Medium

Status Not processed

```
22 int b = 0;
23 int ret;
24
```

25 a = rand();

```
26 if ((5 < a) && (10 < a)) /*Tool should detect this line as error*/ /*ERROR:Redundant
condition*/
27 {
28     b += a;
```

Trace

```
a = rand()
```

01.w_Defects/redundant_cond.c:25

```
22 int b = 0;
23 int ret;
24
```

```
25 a = rand();
```

```
26 if ((5 < a) && (10 < a)) /*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
27 {  
28     b += a;
```

```
a = rand()
```

01.w_Defects/redundant_cond.c:25

```
22 int b = 0;  
23 int ret;  
24
```

```
25 a = rand();
```

```
26 if ((5 < a) && (10 < a)) /*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
27 {  
28     b += a;
```

01.w_Defects/redundant_cond.c:44

Level Medium

Status Not processed

```
41 int b = 0;  
42 int ret;  
43
```

```
44 a = rand();
```

```
45 if ((a < 5) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:Redundant  
condition*/  
46 {  
47     b += a;
```

Trace

```
a = rand()
```

01.w_Defects/redundant_cond.c:44

```
41 int b = 0;  
42 int ret;  
43
```

```
44 a = rand();
```

45 if ((a < 5) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:

Redundant condition*/

```
46 {  
47     b += a;
```

```
a = rand()
```

01.w_Defects/redundant_cond.c:44

```
41 int b = 0;  
42 int ret;  
43
```

```
44 a = rand();
```

45 if ((a < 5) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:

Redundant condition*/

```
46 {  
47     b += a;
```

01.w_Defects/redundant_cond.c:63

Level Medium

Status Not processed

```
60 int b = 0;  
61 int ret;  
62
```

```
63 a = rand();
```

```
64 if (((0 < a) && (a < 10)) && ((2 < a) && (a < 8)))/*Tool should detect this line as error*/
/*ERROR:Redundant condition*/
65 {
66     b += a;
```

Trace

```
a = rand()
```

01.w_Defects/redundant_cond.c:63

```
60 int b = 0;
61 int ret;
62
```

```
63 a = rand();
```

```
64 if (((0 < a) && (a < 10)) && ((2 < a) && (a < 8)))/*Tool should detect this line as
error*/ /*ERROR:Redundant condition*/
65 {
66     b += a;
```

```
a = rand()
```

01.w_Defects/redundant_cond.c:63

```
60 int b = 0;
61 int ret;
62
```

```
63 a = rand();
```

```
64 if (((0 < a) && (a < 10)) && ((2 < a) && (a < 8)))/*Tool should detect this line as
error*/ /*ERROR:Redundant condition*/
65 {
66     b += a;
```

01.w_Defects/redundant_cond.c:82

Level Medium

Status Not processed

```
79 int b = 0;  
80 int ret;  
81  
  
82 a = rand();  
  
83 if (((0 < a) && (a < 8)) && ((5 < a) && (a < 10)))/*Tool should detect this line as error*/  
/*ERROR:Redundant condition*/  
84 {  
85   b += a;
```

Trace

```
a = rand()
```

```
01.w_Defects/redundant_cond.c:82
```

```
79 int b = 0;  
80 int ret;  
81
```

```
82 a = rand();
```

```
83 if (((0 < a) && (a < 8)) && ((5 < a) && (a < 10)))/*Tool should detect this line as  
error*/ /*ERROR:Redundant condition*/  
84 {  
85   b += a;
```

```
a = rand()
```

```
01.w_Defects/redundant_cond.c:82
```

```
79 int b = 0;  
80 int ret;  
81
```

```
82 a = rand();
```

```
83 if (((0 < a) && (a < 8)) && ((5 < a) && (a < 10)))/*Tool should detect this line as  
error*/ /*ERROR:Redundant condition*/  
84 {
```

```
85 b += a;
```

01.w_Defects/redundant_cond.c:101

Level Medium

Status Not processed

```
98 int b = 0;  
99 int ret;  
100
```

101 a = rand();

```
102 if ((5 < a) || (10 < a))/*Tool should detect this line as error*/ /*ERROR:Redundant  
condition*/
```

```
103 {  
104 b += a;
```

Trace

a = rand()

01.w_Defects/redundant_cond.c:101

```
98 int b = 0;  
99 int ret;  
100
```

101 a = rand();

```
102 if ((5 < a) || (10 < a))/*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/
```

```
103 {  
104 b += a;
```

```
a = rand()
```

01.w_Defects/redundant_cond.c:101

```
98 int b = 0;  
99 int ret;  
100
```

```
101 a = rand();
```

```
102 if ((5 < a) || (10 < a))/*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/
```

```
103 {  
104   b += a;
```

01.w_Defects/redundant_cond.c:120

Level Medium

Status Not processed

```
117 int b = 0;  
118 int ret;  
119
```

```
120 a = rand();
```

```
121 if (a < 5)
```

```
122 {
```

```
123   if (a < 10)/*Tool should detect this line as error*/ /*ERROR:Redundant condition*/
```

Trace

```
a = rand()
```

01.w_Defects/redundant_cond.c:120

```
117 int b = 0;  
118 int ret;  
119
```

```
120 a = rand();  
  
121 if (a < 5)  
122 {  
123   if (a < 10)/*Tool should detect this line as error*/ /*ERROR:Redundant  
condition*/
```

a = rand()

01.w_Defects/redundant_cond.c:120

```
117 int b = 0;  
118 int ret;  
119
```

120 a = rand();

```
121 if (a < 5)  
122 {  
123   if (a < 10)/*Tool should detect this line as error*/ /*ERROR:Redundant  
condition*/
```

01.w_Defects/redundant_cond.c:142

Level Medium

Status Not processed

```
139 int b;  
140 int ret;  
141
```

142 a = rand();

```
143 b = ((5 < a) && (10 < a)) ? 0 : 1; /*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
144 ret = b;  
145   sink = ret;
```

Trace

```
a = rand()
```

01.w_Defects/redundant_cond.c:142

```
139 int b;  
140 int ret;  
141
```

```
142 a = rand();
```

```
143 b = ((5 < a) && (10 < a)) ? 0 : 1; /*Tool should detect this line as error*/  
/*ERROR:Redundant condition*/  
144 ret = b;  
145     sink = ret;
```

```
a = rand()
```

01.w_Defects/redundant_cond.c:142

```
139 int b;  
140 int ret;  
141
```

```
142 a = rand();
```

```
143 b = ((5 < a) && (10 < a)) ? 0 : 1; /*Tool should detect this line as error*/  
/*ERROR:Redundant condition*/  
144 ret = b;  
145     sink = ret;
```

01.w_Defects/redundant_cond.c:176

Level Medium

Status Not processed

```
173 int b = 0;  
174 int ret;  
175
```

```
176 a = rand();
```

```
177 while ((5 < a) && (10 < a))/*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
178 {  
179   b += a;
```

Trace

```
a = rand()
```

01.w_Defects/redundant_cond.c:176

```
173 int b = 0;  
174 int ret;  
175
```

176 a = rand();

177 while ((5 < a) && (10 < a))/*Tool should detect this line as error*/ /*ERROR:
Redundant condition*/

```
178 {  
179   b += a;
```

```
a = rand()
```

01.w_Defects/redundant_cond.c:176

```
173 int b = 0;  
174 int ret;  
175
```

176 a = rand();

177 while ((5 < a) && (10 < a))/*Tool should detect this line as error*/ /*ERROR:
Redundant condition*/

```
178 {  
179   b += a;
```

01.w_Defects/redundant_cond.c:196

Level Medium

Status Not processed

```
193 int b = 0;  
194 int ret;  
195  
  
196 a = rand();  
  
197 while ((a < 5) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
198 {  
199   b += a;
```

Trace

```
a = rand()
```

```
01.w_Defects/redundant_cond.c:196
```

```
193 int b = 0;  
194 int ret;  
195
```

```
196 a = rand();
```

```
197 while ((a < 5) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
198 {  
199   b += a;
```

```
a = rand()
```

```
01.w_Defects/redundant_cond.c:196
```

```
193 int b = 0;  
194 int ret;  
195
```

```
196 a = rand();
```

```
197 while ((a < 5) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
198 {
```

```
199 b += a;
```

01.w_Defects/redundant_cond.c:216

Level Medium

Status Not processed

```
213 int b = 0;  
214 int ret;  
215
```

216 a = rand();

```
217 while (((0 < a) && (a < 10)) && ((2 < a) && (a < 8)))/*Tool should detect this line as  
error*/ /*ERROR:Redundant condition*/  
218 {  
219 b += a;
```

Trace

a = rand()

01.w_Defects/redundant_cond.c:216

```
213 int b = 0;  
214 int ret;  
215
```

216 a = rand();

```
217 while (((0 < a) && (a < 10)) && ((2 < a) && (a < 8)))/*Tool should detect this  
line as error*/ /*ERROR:Redundant condition*/  
218 {  
219 b += a;
```

```
a = rand()
```

01.w_Defects/redundant_cond.c:216

```
213 int b = 0;  
214 int ret;  
215
```

```
216 a = rand();
```

```
217 while (((0 < a) && (a < 10)) && ((2 < a) && (a < 8))/*Tool should detect this  
line as error*/ /*ERROR:Redundant condition*/
```

```
218 {  
219   b += a;
```

01.w_Defects/redundant_cond.c:236

Level Medium

Status Not processed

```
233 int b = 0;  
234 int ret;  
235
```

```
236 a = rand();
```

```
237 while (((0 < a) && (a < 8)) && ((5 < a) && (a < 10))/*Tool should detect this line as  
error*/ /*ERROR:Redundant condition*/
```

```
238 {  
239   b += a;
```

Trace

```
a = rand()
```

01.w_Defects/redundant_cond.c:236

```
233 int b = 0;  
234 int ret;  
235
```

236 a = rand();

237 while (((0 < a) && (a < 8)) && ((5 < a) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:Redundant condition*/

238 {

239 b += a;

a = rand()

01.w_Defects/redundant_cond.c:236

233 int b = 0;

234 int ret;

235

236 a = rand();

237 while (((0 < a) && (a < 8)) && ((5 < a) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:Redundant condition*/

238 {

239 b += a;

01.w_Defects/redundant_cond.c:256

Level Medium

Status Not processed

253 int b = 0;

254 int ret;

255

256 a = rand();

257 while ((5 < a) || (10 < a))/*Tool should detect this line as error*/ /*ERROR:Redundant condition*/

258 {

259 b += a;

Trace

```
a = rand()
```

01.w_Defects/redundant_cond.c:256

```
253 int b = 0;  
254 int ret;  
255
```

```
256 a = rand();
```

```
257 while ((5 < a) || (10 < a))/*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
258 {  
259   b += a;
```

```
a = rand()
```

01.w_Defects/redundant_cond.c:256

```
253 int b = 0;  
254 int ret;  
255
```

```
256 a = rand();
```

```
257 while ((5 < a) || (10 < a))/*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
258 {  
259   b += a;
```

01.w_Defects/redundant_cond.c:276

Level Medium

Status Not processed

```
273 int b = 0;  
274 int ret;  
275
```

```
276 a = rand();
```

```
277 do  
278 {  
279   b += a;
```

Trace

```
a = rand()
```

01.w_Defects/redundant_cond.c:276

```
273 int b = 0;  
274 int ret;  
275
```

```
276 a = rand();
```

```
277 do  
278 {  
279   b += a;
```

```
a = rand()
```

01.w_Defects/redundant_cond.c:276

```
273 int b = 0;  
274 int ret;  
275
```

```
276 a = rand();
```

```
277 do  
278 {  
279   b += a;
```

01.w_Defects/sign_conv.c:165

Level Medium

Status Not processed

```
162
163 /*      0 rand() 2147483647 RAND_MAX */
164 /* 1073741823 rand() 1073741823 1073741824 */ */

165 a = rand() - 1073741823;

166
167 ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned
cast */
168     sink = ret;
```

Trace

```
a = rand() - 1073741823
```

01.w_Defects/sign_conv.c:165

```
162
163 /*      0 rand() 2147483647 RAND_MAX */
164 /* 1073741823 rand() 1073741823 1073741824 */ */

165 a = rand() - 1073741823;
```

```
166
167 ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */
168     sink = ret;
```

```
a = rand() - 1073741823
```

01.w_Defects/sign_conv.c:165

```
162
163 /*      0 rand() 2147483647 RAND_MAX */
164 /* 1073741823 rand() 1073741823 1073741824 */ */

165 a = rand() - 1073741823;
```

```
166
167 ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of
unsigned cast */
168     sink = ret;
```

01.w_Defects/sleep_lock.c:70

Level Medium**Status** Not processed

```
67 {  
68   while (1)  
69   {  
  
70     if (rand())  
  
71     {  
72       sleep_lock_001_tsk_001(NULL);  
73     }  
}
```

Trace

if (rand())

01.w_Defects/sleep_lock.c:70

```
67 {  
68   while (1)  
69   {  
  
70     if (rand())  
  
71     {  
72       sleep_lock_001_tsk_001(NULL);  
73     }  
}
```

if (rand())

01.w_Defects/sleep_lock.c:70

```
67 {  
68   while (1)  
69   {  
  
70     if (rand())  
  
71     {  
72       sleep_lock_001_tsk_001(NULL);  
73     }  
}
```

```
72             sleep_lock_001_tsk_001(NULL);  
73 }
```

01.w_Defects/sleep_lock.c:148

Level Medium

Status Not processed

```
145 {  
146   while (1)  
147 {  
  
148     if (rand())  
  
149     {  
150       sleep_lock_002_tsk_001(NULL);  
151     }  
}
```

Trace

```
if (rand())
```

01.w_Defects/sleep_lock.c:148

```
145 {  
146   while (1)  
147 {  
  
148     if (rand())  
  
149     {  
150       sleep_lock_002_tsk_001(NULL);  
151     }  
}
```

```
if (rand())
```

01.w_Defects/sleep_lock.c:148

```
145 {  
146   while (1)  
147   {  
  
148       if (rand())  
  
149       {  
150           sleep_lock_002_tsk_001(NULL);  
151       }  
}
```

01.w_Defects/sleep_lock.c:202

Level Medium

Status Not processed

```
199 {  
200   while (1)  
201   {  
  
202       if (rand())  
  
203       {  
204           sleep_lock_003_tsk_001(NULL);  
205       }  
}
```

Trace

```
if (rand())
```

01.w_Defects/sleep_lock.c:202

```
199 {  
200   while (1)  
201   {  
  
202       if (rand())  
}
```

```
203     {  
204         sleep_lock_003_tsk_001(NULL);  
205     }
```

```
if (rand())
```

01.w_Defects/sleep_lock.c:202

```
199 {  
200     while (1)  
201     {  
  
202         if (rand())  
  
203         {  
204             sleep_lock_003_tsk_001(NULL);  
205         }
```

01.w_Defects/uninit_memory_access.c:385

Level Medium

Status Not processed

```
382  
383 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001 ()  
384 {
```

```
385 int flag = rand();
```

```
386 uninit_memory_access_014_u_001 *u;  
387 switch (flag)  
388 {
```

Trace

```
int flag = rand()
```

01.w_Defects/uninit_memory_access.c:385

```
382
383 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001
()
384 {

385 int flag = rand();

386 uninit_memory_access_014_u_001 *u;
387 switch (flag)
388 {
```

```
int flag = rand()
```

01.w_Defects/uninit_memory_access.c:385

```
382
383 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001
()
384 {

385 int flag = rand();

386 uninit_memory_access_014_u_001 *u;
387 switch (flag)
388 {
```

01.w_Defects/zero_division.c:153

Level Medium

Status Not processed

```
150 int dividend = 1000;
151 int divisor;
152 int ret;
```

```
153 divisor = rand();
```

```
154 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:division by zero */
155 }
156
```

Trace

```
divisor = rand()
```

01.w_Defects/zero_division.c:153

```
150 int dividend = 1000;
151 int divisor;
152 int ret;
```

```
153 divisor = rand();
```

```
154 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:
division by zero */
155 }
156
```

```
divisor = rand()
```

01.w_Defects/zero_division.c:153

```
150 int dividend = 1000;
151 int divisor;
152 int ret;
```

```
153 divisor = rand();
```

```
154 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:
division by zero */
155 }
156
```

02.wo_Defects/bit_shift.c:120

Level Medium

Status Not processed

```
117 int a = 1;
118 int shift;
119 int ret;

120 shift = rand() % 32;

121 ret = a << shift; /*Tool should not detect this line as error*/ /*NO ERROR:Bit shift
error*/
122     sink = ret;
123 }
```

Trace

```
shift = rand() % 32
```

```
02.wo_Defects/bit_shift.c:120
```

```
117 int a = 1;
118 int shift;
119 int ret;

120 shift = rand() % 32;

121 ret = a << shift; /*Tool should not detect this line as error*/ /*NO ERROR:Bit
shift error*/
122     sink = ret;
123 }
```

```
shift = rand() % 32
```

```
02.wo_Defects/bit_shift.c:120
```

```
117 int a = 1;
118 int shift;
119 int ret;

120 shift = rand() % 32;

121 ret = a << shift; /*Tool should not detect this line as error*/ /*NO ERROR:Bit
shift error*/
122     sink = ret;
```

```
123 }
```

02.wo_Defects/conflicting_cond.c:24

Level Medium

Status Not processed

```
21 int b = 0;  
22 int ret;  
23
```

24 a = rand();

```
25 if ((a == 0) || (a == 1)) /*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/  
26 {  
27     b += a;
```

Trace

a = rand()

02.wo_Defects/conflicting_cond.c:24

```
21 int b = 0;  
22 int ret;  
23
```

24 a = rand();

```
25 if ((a == 0) || (a == 1)) /*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/  
26 {  
27     b += a;
```

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:24

```
21 int b = 0;  
22 int ret;  
23
```

```
24 a = rand();
```

```
25 if ((a == 0) || (a == 1)) /*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/  
26 {  
27     b += a;
```

02.wo_Defects/conflicting_cond.c:43

Level Medium

Status Not processed

```
40 int b = 0;  
41 int ret;  
42
```

```
43 a = rand();
```

```
44 if (! ((a < 5) || (10 < a))) /*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/
```

```
45 {  
46     b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:43

```
40 int b = 0;  
41 int ret;  
42
```

```
43 a = rand();  
  
44 if (! ((a < 5) || (10 < a))) /*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/  
45 {  
46     b += a;
```

a = rand()

02.wo_Defects/conflicting_cond.c:43

```
40 int b = 0;  
41 int ret;  
42
```

43 a = rand();

```
44 if (! ((a < 5) || (10 < a))) /*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/  
45 {  
46     b += a;
```

02.wo_Defects/conflicting_cond.c:62

Level Medium

Status Not processed

```
59 int b = 0;  
60 int ret;  
61
```

62 a = rand();

```
63 if (((0 < a) && (a < 2)) || ((8 < a) && (a < 10)))/*Tool should not detect this line as  
error*/ /*No ERROR:contradict condition*/  
64 {  
65     b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:62

```
59 int b = 0;  
60 int ret;  
61
```

```
62 a = rand();
```

```
63 if (((0 < a) && (a < 2)) || ((8 < a) && (a < 10)))/*Tool should not detect this line  
as error*/ /*No ERROR:contradict condition*/
```

```
64 {  
65     b += a;
```

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:62

```
59 int b = 0;  
60 int ret;  
61
```

```
62 a = rand();
```

```
63 if (((0 < a) && (a < 2)) || ((8 < a) && (a < 10)))/*Tool should not detect this line  
as error*/ /*No ERROR:contradict condition*/
```

```
64 {  
65     b += a;
```

02.wo_Defects/conflicting_cond.c:81

Level Medium

Status Not processed

```
78 int b = 0;  
79 int ret;  
80
```

```
81 a = rand();
```

```
82 if (a < 5)
83 {
84   a += 10;
```

Trace

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:81

```
78 int b = 0;
79 int ret;
80

81 a = rand();
```

```
82 if (a < 5)
83 {
84   a += 10;
```

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:81

```
78 int b = 0;
79 int ret;
80

81 a = rand();
```

```
82 if (a < 5)
83 {
84   a += 10;
```

02.wo_Defects/conflicting_cond.c:104

Level Medium

Status Not processed

```
101 int b;  
102 int ret;  
103  
  
104 a = rand();  
  
105 b = ((a == 0) || (a == 1)) ? 0 : 1; /*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/  
106 ret = b;  
107     sink = ret;
```

Trace

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:104

```
101 int b;  
102 int ret;  
103
```

```
104 a = rand();
```

```
105 b = ((a == 0) || (a == 1)) ? 0 : 1; /*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/  
106 ret = b;  
107     sink = ret;
```

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:104

```
101 int b;  
102 int ret;  
103
```

```
104 a = rand();
```

```
105 b = ((a == 0) || (a == 1)) ? 0 : 1; /*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/  
106 ret = b;  
107     sink = ret;
```

02.wo_Defects/conflicting_cond.c:138

Level Medium**Status** Not processed

```
135 int b = 0;  
136 int ret;  
137
```

```
138 a = rand();
```

```
139 while ((a == 0) || (a == 1))/*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/
```

```
140 {  
141   b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:138

```
135 int b = 0;  
136 int ret;  
137
```

```
138 a = rand();
```

```
139 while ((a == 0) || (a == 1))/*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/
```

```
140 {  
141   b += a;
```

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:138

```
135 int b = 0;  
136 int ret;  
137
```

```
138 a = rand();
```

```
139 while ((a == 0) || (a == 1))/*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/  
140 {  
141   b += a;
```

02.wo_Defects/conflicting_cond.c:158

Level Medium

Status Not processed

```
155 int b = 0;  
156 int ret;  
157
```

158 a = rand();

```
159 while (! ((a < 5) || (10 < a)))/*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/  
160 {  
161   b += a;
```

Trace

a = rand()

02.wo_Defects/conflicting_cond.c:158

```
155 int b = 0;  
156 int ret;  
157
```

158 a = rand();

```
159 while (! ((a < 5) || (10 < a)))/*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/  
160 {  
161   b += a;
```

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:158

```
155 int b = 0;  
156 int ret;  
157
```

```
158 a = rand();
```

```
159 while (!(a < 5) || (10 < a))/*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/
```

```
160 {  
161   b += a;
```

02.wo_Defects/conflicting_cond.c:178

Level Medium

Status Not processed

```
175 int b = 0;  
176 int ret;  
177
```

```
178 a = rand();
```

```
179 while (((0 < a) && (a < 2)) || ((8 < a) && (a < 10)))/*Tool should not detect this line as  
error*/ /*No ERROR:contradict condition*/
```

```
180 {  
181   b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:178

```
175 int b = 0;  
176 int ret;  
177
```

178 a = rand();

179 while (((0 < a) && (a < 2)) || ((8 < a) && (a < 10)))/*Tool should not detect this
line as error*/ /*No ERROR:contradict condition*/
180 {
181 b += a;

a = rand()

02.wo_Defects/conflicting_cond.c:178

175 int b = 0;
176 int ret;
177

178 a = rand();

179 while (((0 < a) && (a < 2)) || ((8 < a) && (a < 10)))/*Tool should not detect this
line as error*/ /*No ERROR:contradict condition*/
180 {
181 b += a;

02.wo_Defects/conflicting_cond.c:199

Level Medium

Status Not processed

196
197 do
198 {

199 a = rand();

200 }
201 while ((a == 0) || (a == 1));/*Tool should not detect this line as error*/ /*No ERROR:
contradict condition*/
202 ret = a;

Trace

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:199

```
196
197 do
198 {

199 a = rand();

200 }
201 while ((a == 0) || (a == 1));/*Tool should not detect this line as error*/ /*No
ERROR:contradict condition*/
202 ret = a;
```

```
a = rand()
```

02.wo_Defects/conflicting_cond.c:199

```
196
197 do
198 {

199 a = rand();

200 }
201 while ((a == 0) || (a == 1));/*Tool should not detect this line as error*/ /*No
ERROR:contradict condition*/
202 ret = a;
```

02.wo_Defects/data_lost.c:160

Level Medium

Status Not processed

```
157 {
158 short ret;
159 int a;
```

```
160 a = rand() % 0x8000;
```

```
161 ret = a; /*Tool should not detect this line as error*/ /*No ERROR:Integer precision  
lost because of cast*/  
162     sink = ret;  
163 }
```

Trace

```
a = rand() % 0x8000
```

02.wo_Defects/data_lost.c:160

```
157 {  
158 short ret;  
159 int a;  
  
160 a = rand() % 0x8000;
```

```
161 ret = a; /*Tool should not detect this line as error*/ /*No ERROR:Integer  
precision lost because of cast*/  
162     sink = ret;  
163 }
```

```
a = rand() % 0x8000
```

02.wo_Defects/data_lost.c:160

```
157 {  
158 short ret;  
159 int a;  
  
160 a = rand() % 0x8000;
```

```
161 ret = a; /*Tool should not detect this line as error*/ /*No ERROR:Integer  
precision lost because of cast*/  
162     sink = ret;  
163 }
```

02.wo_Defects/data_overflow.c:205

Level Medium

Status Not processed

```
202 int max = 0x7fffffff;
203 int d;
204 int ret;

205 d = rand() % 2;

206 ret = max + d; /*Tool should not detect this line as error*/ /*No ERROR:Data
Overflow*/
207     sink = ret;
208 }
```

Trace

```
d = rand() % 2
```

```
02.wo_Defects/data_overflow.c:205
```

```
202 int max = 0x7fffffff;
203 int d;
204 int ret;
```

```
205 d = rand() % 2;
```

```
206 ret = max + d; /*Tool should not detect this line as error*/ /*No ERROR:Data
Overflow*/
207     sink = ret;
208 }
```

```
d = rand() % 2
```

```
02.wo_Defects/data_overflow.c:205
```

```
202 int max = 0x7fffffff;
203 int d;
204 int ret;
```

```
205 d = rand() % 2;
```

```
206 ret = max + d; /*Tool should not detect this line as error*/ /*No ERROR:Data
Overflow*/
207     sink = ret;
```

```
208 }
```

02.wo_Defects/double_release.c:53

Level Medium

Status Not processed

```
50 {  
51   while (1)  
52 {  
  
53       if (rand ())  
  
54     {  
55         double_release_001_tsk_001 (NULL);  
56     }
```

Trace

```
if (rand ())
```

02.wo_Defects/double_release.c:53

```
50 {  
51   while (1)  
52 {  
  
53       if (rand ())  
  
54     {  
55         double_release_001_tsk_001 (NULL);  
56     }
```

```
if (rand ())
```

02.wo_Defects/double_release.c:53

```
50 {  
51   while (1)  
52 {  
  
53     if (rand ())  
  
54     {  
55       double_release_001_tsk_001 (NULL);  
56     }
```

02.wo_Defects/double_release.c:106

Level Medium

Status Not processed

```
103 {  
104   while (1)  
105 {  
  
106   if (rand ())  
  
107   {  
108     double_release_002_tsk_001 (NULL);  
109   }
```

Trace

```
if (rand ())
```

02.wo_Defects/double_release.c:106

```
103 {  
104   while (1)  
105 {  
  
106   if (rand ())
```

```
107      {  
108          double_release_002_tsk_001 (NULL);  
109      }
```

```
if (rand ())
```

02.wo_Defects/double_release.c:106

```
103 {  
104 while (1)  
105 {  
  
106     if (rand ())  
  
107     {  
108         double_release_002_tsk_001 (NULL);  
109     }
```

02.wo_Defects/double_release.c:154

Level Medium

Status Not processed

```
151 {  
152 while (1)  
153 {  
  
154     if (rand ())  
  
155     {  
156         double_release_003_tsk_001 (NULL);  
157     }
```

Trace

```
if (rand ())
```

02.wo_Defects/double_release.c:154

```
151 {  
152 while (1)  
153 {  
  
154     if (rand ())  
  
155     {  
156         double_release_003_tsk_001 (NULL);  
157     }
```

```
if (rand ())
```

02.wo_Defects/double_release.c:154

```
151 {  
152 while (1)  
153 {  
  
154     if (rand ())  
  
155     {  
156         double_release_003_tsk_001 (NULL);  
157     }
```

02.wo_Defects/double_release.c:201

Level Medium

Status Not processed

```
198 {  
199 while (1)  
200 {  
  
201     if (rand ())  
  
202     {
```

```
203             double_release_004_tsk_001 (NULL);  
204 }
```

Trace

```
if (rand ())
```

02.wo_Defects/double_release.c:201

```
198 {  
199 while (1)  
200 {  
  
201     if (rand ())  
  
202     {  
203         double_release_004_tsk_001 (NULL);  
204     }
```

```
if (rand ())
```

02.wo_Defects/double_release.c:201

```
198 {  
199 while (1)  
200 {  
  
201     if (rand ())  
  
202     {  
203         double_release_004_tsk_001 (NULL);  
204     }
```

02.wo_Defects/double_release.c:250

Level Medium

Status Not processed

```
247 {
```

```
248 while (1)
249 {
250     if (rand ())
251     {
252         double_release_005_tsk_001 (NULL);
253     }
```

Trace

```
if (rand ())
```

02.wo_Defects/double_release.c:250

```
247 {
248 while (1)
249 {
250     if (rand ())
251     {
252         double_release_005_tsk_001 (NULL);
253     }
```

```
if (rand ())
```

02.wo_Defects/double_release.c:250

```
247 {
248 while (1)
249 {
250     if (rand ())
251     {
252         double_release_005_tsk_001 (NULL);
253     }
```

02.wo_Defects/double_release.c:292

Level Medium**Status** Not processed

```
289 {  
290   while (1)  
291   {  
  
292       if (rand ())  
  
293       {  
294           double_release_006_tsk_001 (NULL);  
295       }  
}
```

Trace

if (rand ())

02.wo_Defects/double_release.c:292

```
289 {  
290   while (1)  
291   {  
  
292       if (rand ())  
  
293       {  
294           double_release_006_tsk_001 (NULL);  
295       }  
}
```

if (rand ())

02.wo_Defects/double_release.c:292

```
289 {  
290   while (1)  
291   {  
  
292       if (rand ())  
  
293       {  
}
```

```
294         double_release_006_tsk_001 (NULL);  
295     }
```

02.wo_Defects/free_null_pointer.c:364

Level Medium

Status Not processed

```
361 static free_null_pointer_011_u_001 *u;  
362 free_null_pointer_011_u_001 * free_null_pointer_011_func_001 ()  
363 {
```

364 int flag = rand();

```
365 flag = 1;  
366 switch (flag)  
367 {
```

Trace

int flag = rand()

02.wo_Defects/free_null_pointer.c:364

```
361 static free_null_pointer_011_u_001 *u;  
362 free_null_pointer_011_u_001 * free_null_pointer_011_func_001 ()  
363 {
```

364 int flag = rand();

```
365 flag = 1;  
366 switch (flag)  
367 {
```

```
int flag = rand()
```

02.wo_Defects/free_null_pointer.c:364

```
361 static free_null_pointer_011_u_001 *u;
362 free_null_pointer_011_u_001 * free_null_pointer_011_func_001 ()
363 {

364 int flag = rand();

365 flag = 1;
366 switch (flag)
367 {
```

02.wo_Defects/func_pointer.c:341

Level Medium

Status Not processed

```
338
339 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
340 {

341 int flag = rand();

342 flag = 1;
343 func_pointer_009_u_001 *u;
344 switch (flag)
```

Trace

```
int flag = rand()
```

02.wo_Defects/func_pointer.c:341

```
338
339 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
340 {

341 int flag = rand();
```

```
342 flag = 1;
343 func_pointer_009_u_001 *u;
344 switch (flag)
```

```
int flag = rand()
```

02.wo_Defects/func_pointer.c:341

```
338
339 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
340 {
```

```
341 int flag = rand();
```

```
342 flag = 1;
343 func_pointer_009_u_001 *u;
344 switch (flag)
```

02.wo_Defects/insign_code.c:23

Level Medium

Status Not processed

```
20 int i;
21 int j;
22
```

```
23 i = rand();
```

```
24 j = i - 1;
25 i = j - 1; /*Tool should not detect this line as error*/ /*No ERROR:Useless Assignment*/
*/
26 printf("%d",i);
```

Trace

```
i = rand()
```

02.wo_Defects/insign_code.c:23

```
20 int i;  
21 int j;  
22  
  
23 i = rand();  
  
24 j = i - 1;  
25 i = j - 1; /*Tool should not detect this line as error*/ /*No ERROR:Useless  
Assignment */  
26 printf("%d",i);
```

```
i = rand()
```

02.wo_Defects/insign_code.c:23

```
20 int i;  
21 int j;  
22  
  
23 i = rand();  
  
24 j = i - 1;  
25 i = j - 1; /*Tool should not detect this line as error*/ /*No ERROR:Useless  
Assignment */  
26 printf("%d",i);
```

02.wo_Defects/memory_leak.c:200

Level Medium

Status Not processed

```
197 void memory_leak_007 ()  
198 {  
199   int ret;  
  
200   ret = memory_leak_007_func_001 (rand());
```

```
201 if(ret >=0 )
202     if(vptr!=NULL)
203     {
```

Trace

```
ret = memory_leak_007_func_001 (rand())
```

02.wo_Defects/memory_leak.c:200

```
197 void memory_leak_007 ()
198 {
199     int ret;

200     ret = memory_leak_007_func_001 (rand());

201     if(ret >=0 )
202         if(vptr!=NULL)
203         {
```

```
ret = memory_leak_007_func_001 (rand())
```

02.wo_Defects/memory_leak.c:200

```
197 void memory_leak_007 ()
198 {
199     int ret;

200     ret = memory_leak_007_func_001 (rand());

201     if(ret >=0 )
202         if(vptr!=NULL)
203         {
```

02.wo_Defects/not_return.c:34

Level Medium

Status Not processed

```
31 void not_return_001 ()  
32 {  
33     int ret;  
  
34     ret = not_return_001_func_001(rand());  
  
35     sink = ret;  
36 }  
37
```

Trace

```
ret = not_return_001_func_001(rand())
```

02.wo_Defects/not_return.c:34

```
31 void not_return_001 ()  
32 {  
33     int ret;  
  
34     ret = not_return_001_func_001(rand());  
  
35     sink = ret;  
36 }  
37
```

```
ret = not_return_001_func_001(rand())
```

02.wo_Defects/not_return.c:34

```
31 void not_return_001 ()  
32 {  
33     int ret;  
  
34     ret = not_return_001_func_001(rand());  
  
35     sink = ret;  
36 }  
37
```

02.wo_Defects/not_return.c:61

Level Medium**Status** Not processed

```
58 void not_return_002 ()  
59 {  
60     int ret;  
  
61     ret = not_return_002_func_001(rand(), rand());  
  
62     sink = ret;  
63 }  
64
```

Trace

```
ret = not_return_002_func_001(rand(),  
rand())
```

02.wo_Defects/not_return.c:61

```
58 void not_return_002 ()  
59 {  
60     int ret;  
  
61     ret = not_return_002_func_001(rand(), rand());  
  
62     sink = ret;  
63 }  
64
```

```
ret = not_return_002_func_001(rand(),  
rand())
```

02.wo_Defects/not_return.c:61

```
58 void not_return_002 ()  
59 {  
60     int ret;  
  
61     ret = not_return_002_func_001(rand(), rand());
```

```
62     sink = ret;
63 }
64
```

02.wo_Defects/not_return.c:87

Level Medium

Status Not processed

```
84 void not_return_003 ()
85 {
86     int ret;

87     ret = not_return_003_func_001(rand());

88     sink = ret;
89 }
90
```

Trace

```
ret = not_return_003_func_001(rand())
```

02.wo_Defects/not_return.c:87

```
84 void not_return_003 ()
85 {
86     int ret;

87     ret = not_return_003_func_001(rand());

88     sink = ret;
89 }
90
```

```
ret = not_return_003_func_001(rand())
```

02.wo_Defects/not_return.c:87

```
84 void not_return_003 ()  
85 {  
86     int ret;  
  
87     ret = not_return_003_func_001(rand());  
  
88     sink = ret;  
89 }  
90
```

02.wo_Defects/not_return.c:111

Level Medium

Status Not processed

```
108 void not_return_004 ()  
109 {  
110     int ret;  
  
111     ret = not_return_004_func_001(rand());  
  
112     sink = ret;  
113 }  
114
```

Trace

```
ret = not_return_004_func_001(rand())
```

02.wo_Defects/not_return.c:111

```
108 void not_return_004 ()  
109 {  
110     int ret;  
  
111     ret = not_return_004_func_001(rand());
```

```
112     sink = ret;
113 }
114
```

```
ret = not_return_004_func_001(rand())
```

02.wo_Defects/not_return.c:111

```
108 void not_return_004 ()
109 {
110     int ret;

111     ret = not_return_004_func_001(rand());

112     sink = ret;
113 }
114
```

02.wo_Defects/overrun_st.c:183

Level Medium

Status Not processed

```
180 {
181     int buf[5];
182     int index;

183     index = rand() % 5;

184     buf[index] = 1; /*Tool should not detect this line as error*/ /*No ERROR: buffer
overrun */
185     sink = buf[idx];
186 }
```

Trace

```
index = rand() % 5
```

02.wo_Defects/overrun_st.c:183

```
180 {  
181   int buf[5];  
182   int index;  
  
183   index = rand() % 5;  
  
184   buf[index] = 1; /*Tool should not detect this line as error*/ /*No ERROR:  
buffer overrun */  
185     sink = buf[idx];  
186 }
```

```
index = rand() % 5
```

02.wo_Defects/overrun_st.c:183

```
180 {  
181   int buf[5];  
182   int index;  
  
183   index = rand() % 5;  
  
184   buf[index] = 1; /*Tool should not detect this line as error*/ /*No ERROR:  
buffer overrun */  
185     sink = buf[idx];  
186 }
```

02.wo_Defects/overrun_st.c:443

Level Medium

Status Not processed

```
440   int *p;  
441   int index;  
442   p = buf;
```

```
443   index = rand() % 5;
```

```
444 *(p + index) = 1; /*Tool should not detect this line as error*/ /*No ERROR: buffer
overrun */
445 }
446
```

Trace

```
index = rand() % 5
```

02.wo_Defects/overrun_st.c:443

```
440 int *p;
441 int index;
442 p = buf;
```

```
443 index = rand() % 5;
```

```
444 *(p + index) = 1; /*Tool should not detect this line as error*/ /*No ERROR:
buffer overrun */
445 }
446
```

```
index = rand() % 5
```

02.wo_Defects/overrun_st.c:443

```
440 int *p;
441 int index;
442 p = buf;
```

```
443 index = rand() % 5;
```

```
444 *(p + index) = 1; /*Tool should not detect this line as error*/ /*No ERROR:
buffer overrun */
445 }
446
```

02.wo_Defects/redundant_cond.c:26

Level Medium

Status Not processed

```
23 int b = 0;  
24 int ret;  
25  
  
26 a = rand();  
  
27 if ( a < 10 ) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
28 {  
29     b += a;
```

Trace

```
a = rand()
```

```
02.wo_Defects/redundant_cond.c:26
```

```
23 int b = 0;  
24 int ret;  
25
```

```
26 a = rand();
```

```
27 if ( a < 10 ) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
28 {  
29     b += a;
```

```
a = rand()
```

```
02.wo_Defects/redundant_cond.c:26
```

```
23 int b = 0;  
24 int ret;  
25
```

```
26 a = rand();
```

```
27 if ( a < 10 ) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
28 {
```

```
29    b += a;
```

02.wo_Defects/redundant_cond.c:45

Level Medium

Status Not processed

```
42 int b = 0;  
43 int ret;  
44
```

45 a = rand();

```
46 if (a < 5) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/
```

```
47 {  
48    b += a;
```

Trace

a = rand()

02.wo_Defects/redundant_cond.c:45

```
42 int b = 0;  
43 int ret;  
44
```

45 a = rand();

```
46 if (a < 5) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/
```

```
47 {  
48    b += a;
```

```
a = rand()
```

02.wo_Defects/redundant_cond.c:45

```
42 int b = 0;  
43 int ret;  
44
```

```
45 a = rand();
```

```
46 if (a < 5) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
47 {  
48     b += a;
```

02.wo_Defects/redundant_cond.c:64

Level Medium

Status Not processed

```
61 int b = 0;  
62 int ret;  
63
```

```
64 a = rand();
```

```
65 if ( a < 10 ) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
66 {  
67     b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/redundant_cond.c:64

```
61 int b = 0;  
62 int ret;  
63
```

```
64 a = rand();  
  
65 if ( a < 10 ) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
66 {  
67   b += a;
```

a = rand()

02.wo_Defects/redundant_cond.c:64

```
61 int b = 0;  
62 int ret;  
63
```

64 a = rand();

```
65 if ( a < 10 ) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
66 {  
67   b += a;
```

02.wo_Defects/redundant_cond.c:83

Level Medium

Status Not processed

```
80 int b = 0;  
81 int ret;  
82
```

83 a = rand();

```
84 if ( a < 10 )/*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
85 {  
86   b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/redundant_cond.c:83

```
80 int b = 0;  
81 int ret;  
82
```

```
83 a = rand();
```

```
84 if ( a < 10 )/*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
85 {  
86   b += a;
```

```
a = rand()
```

02.wo_Defects/redundant_cond.c:83

```
80 int b = 0;  
81 int ret;  
82
```

```
83 a = rand();
```

```
84 if ( a < 10 )/*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
85 {  
86   b += a;
```

02.wo_Defects/redundant_cond.c:102

Level Medium

Status Not processed

```
99 int b = 0;  
100 int ret;  
101
```

```
102 a = rand();
```

```
103 if (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
104 {  
105   b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/redundant_cond.c:102

```
99 int b = 0;  
100 int ret;  
101
```

102 a = rand();

```
103 if (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
104 {  
105   b += a;
```

```
a = rand()
```

02.wo_Defects/redundant_cond.c:102

```
99 int b = 0;  
100 int ret;  
101
```

102 a = rand();

```
103 if (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
104 {  
105   b += a;
```

02.wo_Defects/redundant_cond.c:121

Level Medium

Status Not processed

```
118 int b = 0;  
119 int ret;  
120  
  
121 a = rand();  
  
122 if (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
123 {  
124   b += a;
```

Trace

```
a = rand()
```

```
02.wo_Defects/redundant_cond.c:121
```

```
118 int b = 0;  
119 int ret;  
120
```

```
121 a = rand();
```

```
122 if (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:  
Redundant condition*/
```

```
123 {  
124   b += a;
```

```
a = rand()
```

```
02.wo_Defects/redundant_cond.c:121
```

```
118 int b = 0;  
119 int ret;  
120
```

```
121 a = rand();
```

```
122 if (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:  
Redundant condition*/  
123 {
```

```
124 b += a;
```

02.wo_Defects/redundant_cond.c:141

Level Medium

Status Not processed

```
138 int b;  
139 int ret;  
140
```

141 a = rand();

142 b = ((a < 10)) ? 0 : 1; /*Tool should not detect this line as error*/ /*No ERROR:
Redundant condition*/

```
143 ret = b;  
144     sink = ret;
```

Trace

a = rand()

02.wo_Defects/redundant_cond.c:141

```
138 int b;  
139 int ret;  
140
```

141 a = rand();

142 b = ((a < 10)) ? 0 : 1; /*Tool should not detect this line as error*/ /*No
ERROR:Redundant condition*/

```
143 ret = b;  
144     sink = ret;
```

```
a = rand()
```

02.wo_Defects/redundant_cond.c:141

```
138 int b;  
139 int ret;  
140
```

```
141 a = rand();
```

```
142 b = ((a < 10) ) ? 0 : 1; /*Tool should not detect this line as error*/ /*No  
ERROR:Redundant condition*/  
143 ret = b;  
144     sink = ret;
```

02.wo_Defects/redundant_cond.c:175

Level Medium

Status Not processed

```
172 int b = 0;  
173 int ret;  
174
```

```
175 a = rand();
```

```
176 while (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
177 {  
178     b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/redundant_cond.c:175

```
172 int b = 0;  
173 int ret;  
174
```

```
175 a = rand();
```

```
176 while (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:  
Redundant condition*/  
177 {  
178   b += a;
```

```
a = rand()
```

```
02.wo_Defects/redundant_cond.c:175
```

```
172 int b = 0;  
173 int ret;  
174
```

```
175 a = rand();
```

```
176 while (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:  
Redundant condition*/  
177 {  
178   b += a;
```

```
02.wo_Defects/redundant_cond.c:195
```

Level Medium

Status Not processed

```
192 int b = 0;  
193 int ret;  
194
```

```
195 a = rand();
```

```
196 while ((a < 5)) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
197 {  
198   b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/redundant_cond.c:195

```
192 int b = 0;  
193 int ret;  
194
```

```
195 a = rand();
```

196 while ((a < 5)) /*Tool should not detect this line as error*/ /*No ERROR:
Redundant condition*/

```
197 {  
198   b += a;
```

```
a = rand()
```

02.wo_Defects/redundant_cond.c:195

```
192 int b = 0;  
193 int ret;  
194
```

```
195 a = rand();
```

196 while ((a < 5)) /*Tool should not detect this line as error*/ /*No ERROR:
Redundant condition*/

```
197 {  
198   b += a;
```

02.wo_Defects/redundant_cond.c:215

Level Medium

Status Not processed

```
212 int b = 0;  
213 int ret;  
214
```

```
215 a = rand();
```

```
216 while ((a < 8)) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
217 {  
218   b += a;
```

Trace

```
a = rand()
```

02.wo_Defects/redundant_cond.c:215

```
212 int b = 0;  
213 int ret;  
214
```

215 a = rand();

216 while ((a < 8)) /*Tool should not detect this line as error*/ /*No ERROR:
Redundant condition*/

```
217 {  
218   b += a;
```

```
a = rand()
```

02.wo_Defects/redundant_cond.c:215

```
212 int b = 0;  
213 int ret;  
214
```

215 a = rand();

216 while ((a < 8)) /*Tool should not detect this line as error*/ /*No ERROR:
Redundant condition*/

```
217 {  
218   b += a;
```

02.wo_Defects/redundant_cond.c:235

Level Medium

Status Not processed

```
232 int b = 0;  
233 int ret;  
234
```

```
235 a = rand();
```

```
236 while (a < 10)/*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
237 {  
238   b += a;
```

Trace

```
a = rand()
```

```
02.wo_Defects/redundant_cond.c:235
```

```
232 int b = 0;  
233 int ret;  
234
```

```
235 a = rand();
```

```
236 while (a < 10)/*Tool should not detect this line as error*/ /*No ERROR:  
Redundant condition*/  
237 {  
238   b += a;
```

```
a = rand()
```

```
02.wo_Defects/redundant_cond.c:235
```

```
232 int b = 0;  
233 int ret;  
234
```

```
235 a = rand();
```

```
236 while (a < 10)/*Tool should not detect this line as error*/ /*No ERROR:  
Redundant condition*/  
237 {
```

```
238 b += a;
```

02.wo_Defects/redundant_cond.c:255

Level Medium

Status Not processed

```
252 int b = 0;  
253 int ret;  
254
```

255 a = rand();

```
256 while ((a < 10))/*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
257 {  
258   b += a;
```

Trace

a = rand()

02.wo_Defects/redundant_cond.c:255

```
252 int b = 0;  
253 int ret;  
254
```

255 a = rand();

```
256 while ((a < 10))/*Tool should not detect this line as error*/ /*No ERROR:  
Redundant condition*/  
257 {  
258   b += a;
```

```
a = rand()
```

02.wo_Defects/redundant_cond.c:255

```
252 int b = 0;  
253 int ret;  
254
```

```
255 a = rand();
```

```
256 while ((a < 10))/*Tool should not detect this line as error*/ /*No ERROR:
```

```
Redundant condition*/
```

```
257 {  
258   b += a;
```

02.wo_Defects/redundant_cond.c:275

Level Medium

Status Not processed

```
272 int b = 0;  
273 int ret;  
274
```

```
275 a = rand();
```

```
276 do {  
277   // JDR: cast to unsigned to avoid UB  
278   b += (unsigned)a;
```

Trace

```
a = rand()
```

02.wo_Defects/redundant_cond.c:275

```
272 int b = 0;  
273 int ret;  
274
```

```
275 a = rand();  
  
276 do {  
277 // JDR: cast to unsigned to avoid UB  
278 b += (unsigned)a;
```

```
a = rand()
```

02.wo_Defects/redundant_cond.c:275

```
272 int b = 0;  
273 int ret;  
274
```

```
275 a = rand();
```

```
276 do {  
277 // JDR: cast to unsigned to avoid UB  
278 b += (unsigned)a;
```

02.wo_Defects/sign_conv.c:165

Level Medium

Status Not processed

```
162 unsigned int ret;  
163  
164 /* 0 <= rand() <= 2147483647 (RAND_MAX) */
```

```
165 a = rand();
```

```
166  
167 ret = a; /*Tool should not detect this line as error*/ /*NO ERROR : Integer sign lost  
because of unsigned cast */  
168     sink = ret;
```

Trace

```
a = rand()
```

02.wo_Defects/sign_conv.c:165

```
162 unsigned int ret;
163
164 /* 0 <= rand() <= 2147483647 (RAND_MAX) */

165 a = rand();

166
167 ret = a; /*Tool should not detect this line as error*/ /*NO ERROR : Integer sign
lost because of unsigned cast */
168     sink = ret;
```

```
a = rand()
```

02.wo_Defects/sign_conv.c:165

```
162 unsigned int ret;
163
164 /* 0 <= rand() <= 2147483647 (RAND_MAX) */

165 a = rand();

166
167 ret = a; /*Tool should not detect this line as error*/ /*NO ERROR : Integer sign
lost because of unsigned cast */
168     sink = ret;
```

02.wo_Defects/uninit_memory_access.c:401

Level Medium

Status Not processed

```
398
399 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001 ()
400 {


```

```
401 int flag = rand();
```

```
402 flag = 1;
403 uninit_memory_access_014_u_001 *u;
404 switch (flag)
```

Trace

```
int flag = rand()
```

02.wo_Defects/uninit_memory_access.c:401

```
398
399 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001
()
400 {
```

```
401 int flag = rand();
```

```
402 flag = 1;
403 uninit_memory_access_014_u_001 *u;
404 switch (flag)
```

```
int flag = rand()
```

02.wo_Defects/uninit_memory_access.c:401

```
398
399 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001
()
400 {
```

```
401 int flag = rand();
```

```
402 flag = 1;
403 uninit_memory_access_014_u_001 *u;
404 switch (flag)
```

02.wo_Defects/zero_division.c:151

Level Medium

Status Not processed

```
148 int dividend = 1000;  
149 int divisor;  
150 int ret;  
  
151 divisor = rand();  
  
152 if (divisor != 0)  
153 {  
154   ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */
```

Trace

```
divisor = rand()
```

02.wo_Defects/zero_division.c:151

```
148 int dividend = 1000;  
149 int divisor;  
150 int ret;
```

```
151 divisor = rand();
```

```
152 if (divisor != 0)  
153 {  
154   ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

```
divisor = rand()
```

02.wo_Defects/zero_division.c:151

```
148 int dividend = 1000;  
149 int divisor;  
150 int ret;
```

```
151 divisor = rand();
```

```
152 if (divisor != 0)  
153 {  
154   ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

Use after free (C/C++)

Description

The application is trying to use the freed memory. This may lead to incorrect behavior of the application.

The use of previously-freed memory can have any number of adverse consequences, ranging from the corruption of valid data to the execution of arbitrary code. The simplest way data corruption may occur involves the system's reuse of the freed memory. Use-after-free errors have two common causes:

- Error conditions and other exceptional circumstances.
- Confusion over which part of the program is responsible for freeing the memory.

Example

Incorrect use example:

```
char* ptr = (char*)malloc (SIZE);
if (err) {
    abrt = 1;
    free(ptr);
}
...
if (abrt) {
    logError("operation aborted before commit", ptr);
}
```

Recommendations

- Free memory after using it.

- Do not use a variable after freeing it.

Links

1. What happens when you try to free() already freed memory in c? — stackoverflow.com
2. CWE-416: Use After Free

Vulnerability Entries

01.w_Defects/invalid_memory_access.c:45

Level Medium

Status Not processed

```
42 }
43 }
44 if(flag == 10)
```

45 a = *(ptr+1); /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

```
46
47 }
48
```

Trace

vflag == 1

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {

663     if (vflag == 1 || vflag == 888)

664     {
665         invalid_memory_access_001();
666     }
```

```
*(ptr+1)
```

01.w_Defects/invalid_memory_access.c:45

```
42    }
43    }
44    if(flag == 10)

45        a = *(ptr+1); /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

46
47 }
48
```

01.w_Defects/invalid_memory_access.c:84

Level Medium

Status Not processed

```
81      ;
82  }
83  else
```

```
84        a = *(dptr+1);/*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/
```

```
85  printf("%lf",a);
86 }
87
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
```

```
663 if (vflag == 1 || vflag ==888)

664 {
665     invalid_memory_access_001();
666 }
```

*(dptr+1)

01.w_Defects/invalid_memory_access.c:84

```
81 ;
82 }
83 else

84         a = *(dptr+1);/*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

85     printf("%lf",a);
86 }
87
```

01.w_Defects/invalid_memory_access.c:133

Level Medium

Status Not processed

```
130 }
131 if(staticflag1)
132 {

133     printf("String= %s",buf); /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

134 }
135 }
136
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
663     if (vflag == 1 || vflag == 888)
664     {
665         invalid_memory_access_001();
666     }
}
```

```
printf("String= %s",buf); /*Tool should detect
this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

01.w_Defects/invalid_memory_access.c:133

```
130 }
131 if(staticflag1)
132 {
133     printf("String= %s",buf); /*Tool should detect this line as error*/
/*ERROR:Invalid memory access to already freed area*/
134 }
135 }
136
```

01.w_Defects/invalid_memory_access.c:188

Level Medium

Status Not processed

```
185 free(buf3);
186 free(buf4);
187 free(buf5);
```

```
188 *((*pbuf[1])+1) =buf2[0]; /*Tool should detect this line as error*/ /*ERROR:Invalid
```

memory access to already freed area*/

```
189 }  
190  
191 /*
```

Trace

vflag == 1

```
01.w_Defects/invalid_memory_access.c:663  
660 extern volatile int vflag;  
661 void invalid_memory_access_main ()  
662 {  
  
663 if (vflag == 1 || vflag ==888)  
  
664 {  
665     invalid_memory_access_001();  
666 }
```

buf2[0]

```
01.w_Defects/invalid_memory_access.c:188
```

```
185 free(buf3);  
186 free(buf4);  
187 free(buf5);  
  
188 *((*pbuff[1])+1) =buf2[0]; /*Tool should detect this line as error*/ /*ERROR:  
189 Invalid memory access to already freed area*/  
190  
191 /*
```

01.w_Defects/invalid_memory_access.c:210

Level Medium

Status Not processed

```
207     ptr[i] = NULL;
208 }
209 free(ptr);

210 strcpy(*(ptr+2),"String"); /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

211 }
212 /*
213 /*
```

Trace

vflag == 1

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {

663 if (vflag == 1 || vflag ==888)

664 {
665     invalid_memory_access_001();
666 }
```

strcpy(*(ptr+2),"String")

01.w_Defects/invalid_memory_access.c:210

```
207     ptr[i] = NULL;
208 }
209 free(ptr);

210 strcpy(*(ptr+2),"String"); /*Tool should detect this line as error*/ /*ERROR:
Invalid memory access to already freed area*/

211 }
212 /*
213 /*
```

01.w_Defects/invalid_memory_access.c:224

Level Medium**Status** Not processed

```
221 if(buf != NULL)
222 {
223     free(buf);
```

```
224     memcpy(buf,buf1,11); /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/
```

```
225 }
226 }
227
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {

663     if (vflag == 1 || vflag ==888)

664     {
665         invalid_memory_access_001();
666     }
```

```
memcpy(buf,buf1,11)
```

01.w_Defects/invalid_memory_access.c:224

```
221 if(buf != NULL)
222 {
223 free(buf);

224 memcpy(buf,buf1,11); /*Tool should detect this line as error*/ /*ERROR:
   Invalid memory access to already freed area*/

225 }
226 }
227
```

01.w_Defects/invalid_memory_access.c:270

Level Medium

Status Not processed

```
267 free(u->s1->a);
268 free(u->s1);
269 free(u);
```

270 p->s1->a[0] = 1; /*Tool should detect this line as error*/ /*ERROR:Invalid memory
 access to already freed area*/

```
271 }
272 }
273
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
```

```
663 if (vflag == 1 || vflag ==888)

664 {
665     invalid_memory_access_001();
666 }
```

p->s1

01.w_Defects/invalid_memory_access.c:270

```
267 free(u->s1->a);
268 free(u->s1);
269 free(u);

270 p->s1->a[0] = 1; /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

271 }
272 }
273
```

01.w_Defects/invalid_memory_access.c:294

Level Medium

Status Not processed

```
291     *(ptr1+i) = ptr[i];
292 }
293 free(ptr1);
```

```
294 *(ptr1+1) = ptr[1];/*Tool should detect this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

```
295 }
296
297 /*
```

Trace

vflag == 1

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
663     if (vflag == 1 || vflag == 888)
664     {
665         invalid_memory_access_001();
666     }
```

*(ptr1+1)

01.w_Defects/invalid_memory_access.c:294

```
291         *(ptr1+i) = ptr[i];
292     }
293     free(ptr1);
294     *(ptr1+1) = ptr[1]; /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/
295 }
296 /*
297 /*
```

01.w_Defects/invalid_memory_access.c:320

Level Medium

Status Not processed

```
317     if(j>10)
318     break;
319 }
```

```
320     *(ptr+i) = i; /*Tool should detect this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

```
321 }  
322  
323 /*
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;  
661 void invalid_memory_access_main ()  
662 {  
  
663 if (vflag == 1 || vflag ==888)  
  
664 {  
665     invalid_memory_access_001();  
666 }
```

```
*(ptr+i)
```

01.w_Defects/invalid_memory_access.c:320

```
317     if(j>10)  
318     break;  
319 }  
  
320 *(ptr+i) = i; /*Tool should detect this line as error*/ /*ERROR:Invalid memory  
access to already freed area*/  
  
321 }  
322  
323 /*
```

01.w_Defects/invalid_memory_access.c:371

Level Medium

Status Not processed

```
368         break;  
369     }  
370 }
```

```
371 return (i+s->a);/*Tool should detect this line as error*/ /*ERROR:Invalid memory  
access to already freed area*/
```

```
372 }  
373  
374 void invalid_memory_access_012 ()
```

Trace

```
vflag == 1
```

```
01.w_Defects/invalid_memory_access.c:663
```

```
660 extern volatile int vflag;  
661 void invalid_memory_access_main ()  
662 {  
  
663 if (vflag == 1 || vflag ==888)  
  
664 {  
665     invalid_memory_access_001();  
666 }
```

```
s->a
```

```
01.w_Defects/invalid_memory_access.c:371
```

```
368         break;  
369     }  
370 }  
  
371 return (i+s->a);/*Tool should detect this line as error*/ /*ERROR:Invalid  
memory access to already freed area*/  
  
372 }  
373  
374 void invalid_memory_access_012 ()
```

01.w_Defects/invalid_memory_access.c:432

Level Medium**Status** Not processed

```
429             break;
430         }
431 }
```

432 return invalid_memory_access_013_s_001_s_gbl->a; /*Tool should detect this line
as error*/ /*ERROR:Invalid memory access to already freed area*/

```
433 /*      return i;*/
434 }
435
```

Trace

vflag == 1

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
663     if (vflag == 1 || vflag ==888)
664     {
665         invalid_memory_access_001();
666     }
}
```

invalid_memory_access_013_s_001_s_gbl->a

01.w_Defects/invalid_memory_access.c:432

```
429             break;
430         }
431 }
```

432 return invalid_memory_access_013_s_001_s_gbl->a; /*Tool should detect

```
this line as error*/ /*ERROR:Invalid memory access to already freed area*/
```

```
433 /*      return i;*/
434 }
435
```

01.w_Defects/invalid_memory_access.c:516

Level Medium

Status Not processed

```
513     str_rev[i] = '\0';
514 }
515 free(str_rev);
```

```
516 return str_rev; /*Tool should detect this line as error*/ /*ERROR:Invalid memory
access to already freed area*/
```

```
517 }
518 else
519 {
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
```

```
663 if (vflag == 1 || vflag == 888)
```

```
664 {
665     invalid_memory_access_001();
666 }
```

```
return str_rev; /*Tool should detect this line  
as error*/ /*ERROR:Invalid memory access  
to already freed area*/
```

01.w_Defects/invalid_memory_access.c:516

```
513     str_rev[i] = '\0';  
514 }  
515 free(str_rev);
```

```
516 return str_rev; /*Tool should detect this line as error*/ /*ERROR:Invalid  
memory access to already freed area*/
```

```
517 }  
518 else  
519 {
```

01.w_Defects/invalid_memory_access.c:568

Level Medium

Status Not processed

```
565 void invalid_memory_access_016_func_003()  
566 {  
567     char s[10];
```

```
568     strcpy(s,invalid_memory_access_016_doubleptr_gbl[0]); /*Tool should detect this  
line as error*/ /*ERROR:Invalid memory access to already freed area*/
```

```
569 }  
570  
571 void invalid_memory_access_016()
```

Trace

```
vflag == 1
```

01.w_Defects/invalid_memory_access.c:663

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {
663     if (vflag == 1 || vflag ==888)
664     {
665         invalid_memory_access_001();
666     }
```

```
strcpy(s,
invalid_memory_access_016_doubleptr_gbl
[0])
```

01.w_Defects/invalid_memory_access.c:568

```
565 void invalid_memory_access_016_func_003()
566 {
567     char s[10] ;
568     strcpy(s,invalid_memory_access_016_doubleptr_gbl[0]);/*Tool should detect
this line as error*/ /*ERROR:Invalid memory access to already freed area*/
569 }
570
571 void invalid_memory_access_016()
```

01.w_Defects/invalid_memory_access.c:622

Level Medium

Status Not processed

```
619 void invalid_memory_access_017_func_004()
620 {
621     char s[10] ;
```

```
622     strcpy(s,invalid_memory_access_017_doubleptr_gbl);/*Tool should detect this line
```

```
as error*/ /*ERROR:Invalid memory access to already freed area*/
```

```
623 }
624
625 void invalid_memory_access_017()
```

Trace

```
vflag == 1
```

```
01.w_Defects/invalid_memory_access.c:663
```

```
660 extern volatile int vflag;
661 void invalid_memory_access_main ()
662 {

663 if (vflag == 1 || vflag ==888)

664 {
665     invalid_memory_access_001();
666 }
```

```
strcpy(s,
invalid_memory_access_017_doubleptr_gbl
)
```

```
01.w_Defects/invalid_memory_access.c:622
```

```
619 void invalid_memory_access_017_func_004()
620 {
621     char s[10] ;

622 strcpy(s,invalid_memory_access_017_doubleptr_gbl);/*Tool should detect
this line as error*/ /*ERROR:Invalid memory access to already freed area*/

623 }
624
625 void invalid_memory_access_017()
```

02.wo_Defects/buffer_underrun_dynamic.c:723

Level Medium

Status Not processed

```
720 doubleptr[i]=(char*) malloc(10*sizeof(char));/*Tool should not detect this line as  
error*/ /*No ERROR:Buffer Underrun*/  
721 if(doubleptr[i]!=NULL)  
722 {  
  
723     doubleptr[0][0]='T';  
  
724     free(doubleptr[i]);  
725 }  
726 }
```

Trace

vflag == 1

02.wo_Defects/buffer_underrun_dynamic.c:792

```
789 extern volatile int vflag;  
790 void dynamic_buffer_underrun_main ()  
791 {  
  
792     if (vflag == 1 || vflag ==888)  
  
793     {  
794         dynamic_buffer_underrun_001();  
795     }
```

doubleptr[0][0]

02.wo_Defects/buffer_underrun_dynamic.c:723

```
720 doubleptr[i]=(char*) malloc(10*sizeof(char));/*Tool should not detect this line as  
error*/ /*No ERROR:Buffer Underrun*/  
721 if(doubleptr[i]!=NULL)  
722 {  
  
723     doubleptr[0][0]='T';
```

```
724 free(doubleptr[i]);  
725 }  
726 }
```

Weak random number generator (C/C++)

Description

Used pseudorandom number generator (PRNG) is not secure since it generates predictable sequences. This can be exploited to bypass authentication and hijack the user's session, as well as to carry out the DNS cache poisoning attack.

PRNGs generate number sequences based on the initial value of the seed. There are two types of PRNG: statistical and cryptographic. Statistical PRNGs generate predictable sequences, which are similar to random according to the statistical characteristics. They may not be used for security purposes. The result of the cryptographic PRNG, on the contrary, is impossible to predict if the value of seed is derived from a source with high entropy. The value of the current time has a small entropy and is also insecure as a seed.

Insufficient Cryptography vulnerabilities take the fifth place in the “OWASP Top 10 2016” mobile application vulnerabilities ranking.

Example

In the following example, the application generates a predictable sequence of pseudorandom numbers:

```
#include <stdlib.h>  
int r = rand() % N // in range 0 to N-1
```

The random(), arc4random(), arc4random_uniform(), and srandom() methods are also cryptographically insecure.

It is recommended to use the data from the dev/random file (system entropy source):

```
FILE *fp = fopen("/dev/random", "r");
```

```
if (!fp) {  
    perror("randgetter");  
    exit(-1);
```

```
}
```

```
uint64_t value = 0;
int i;
for (i=0; i<sizeof(value); i++) {
    value <<= 8;
    value |= fgetc(fp);
}
fclose(fp);
```

Recommendations

- Use cryptographic PRNG to generate pseudo-random numbers for information security purposes.
- Use sources of high entropy to generate a seed for PRNG.

Links

1. OWASP: Insecure randomness
2. CWE-330: Use of Insufficiently Random Values
3. CERT: MSC02-J. Generate strong random numbers
4. Generating Random Numbers — developer.apple.com
5. Mobile Top 10 2016-M5-Insufficient Cryptography

Vulnerability Entries

01.w_Defects/bit_shift.c:120

Level Medium

Status Not processed

```
117 int a = 1;
118 int shift;
119 int ret;
```

```
120 shift = rand();
```

```
121 ret = a << shift; /*Tool should detect this line as error*/ /*ERROR:Bit shift error*/
122     sink = ret;
123 }
```

01.w_Defects/buffer_underrun_dynamic.c:249

Level Medium**Status** Not processed

```
246 {  
247   int *buf=(int*) calloc(5,sizeof(int));  
248   int index = 5;  
  
249   index = rand()-2;  
  
250   if(buf!=NULL)  
251   {  
252
```

01.w_Defects/conflicting_cond.c:23

Level Medium**Status** Not processed

```
20 int b = 0;  
21 int ret;  
22  
  
23 a = rand();  
  
24 if ((a == 0) && (a == 1))/*Tool should detect this line as error*/ /*ERROR:contradict  
condition*/  
25 {  
26   b += a;
```

01.w_Defects/conflicting_cond.c:42

Level Medium**Status** Not processed

```
39 int b = 0;  
40 int ret;  
41
```

42 a = rand();

```
43 if ((a < 5) && (10 < a))/*Tool should detect this line as error*/ /*ERROR:contradict  
condition*/  
44 {  
45     b += a;
```

01.w_Defects/conflicting_cond.c:61

Level Medium

Status Not processed

```
58 int b = 0;  
59 int ret;  
60
```

61 a = rand();

```
62 if (((0 < a) && (a < 2)) && ((8 < a) && (a < 10))) /*Tool should detect this line as error*/  
/*ERROR:contradict condition*/  
63 {  
64     b += a;
```

01.w_Defects/conflicting_cond.c:80

Level Medium

Status Not processed

```
77 int b = 0;  
78 int ret;  
79
```

80 a = rand();

```
81 if (a < 5)  
82 {  
83     if (10 < a) /*Tool should detect this line as error*/ /*ERROR:contradict condition*/
```

01.w_Defects/conflicting_cond.c:102

Level Medium**Status** Not processed

```
99 int b;  
100 int ret;  
101
```

102 a = rand();

```
103 b = ((a == 0) && (a == 1)) ? 0 : 1; /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/
```

```
104 ret = b;  
105     sink = ret;
```

01.w_Defects/conflicting_cond.c:136

Level Medium**Status** Not processed

```
133 int b = 0;  
134 int ret;  
135
```

136 a = rand();

```
137 while ((a == 0) && (a == 1)) /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/
```

```
138 {  
139     b += a;
```

01.w_Defects/conflicting_cond.c:156

Level Medium**Status** Not processed

```
153 int b = 0;  
154 int ret;  
155
```

```
156 a = rand();
```

```
157 while ((a < 5) && (10 < a)) /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
158 {  
159   b += a;
```

01.w_Defects/conflicting_cond.c:176

Level Medium

Status Not processed

```
173 int b = 0;  
174 int ret;  
175
```

```
176 a = rand();
```

```
177 while (((0 < a) && (a < 2)) && ((8 < a) && (a < 10))) /*Tool should detect this line as  
error*/ /*ERROR:contradict condition*/  
178 {  
179   b += a;
```

01.w_Defects/conflicting_cond.c:197

Level Medium

Status Not processed

```
194  
195 do  
196 {
```

```
197   a = rand();
```

```
198 }  
199 while ((a == 0) && (a == 1)); /*Tool should detect this line as error*/ /*ERROR:  
contradict condition*/  
200 ret = a;
```

01.w_Defects/data_lost.c:157

Level Medium**Status** Not processed

```
154 {  
155     short ret;  
156     int a;  
  
157     a = rand();  
  
158     ret = a; /*Tool should detect this line as error*/ /*ERROR:Integer precision lost  
because of cast*/  
159     sink = ret;  
160 }
```

01.w_Defects/data_overflow.c:204

Level Medium**Status** Not processed

```
201     int max = 0x7fffffff;  
202     int d;  
203     int ret;  
  
204     d = rand();  
  
205     ret = max + d; /*Tool should detect this line as error*/ /*ERROR:Data Overflow*/  
206     sink = ret;  
207 }
```

01.w_Defects/double_free.c:81

Level Medium**Status** Not processed

```
78     *(ptr+i)='a';
79 }
80

81 if (rand() % 2==0)

82 {
83     free(ptr);
84 }
```

01.w_Defects/double_free.c:86

Level Medium**Status** Not processed

```
83     free(ptr);
84 }
85
```

```
86 if(rand() % 3==0)
```

```
87 free(ptr); /*Tool should detect this line as error*/ /*ERROR:Double free*/
88 }
89
```

01.w_Defects/double_release.c:54

Level Medium**Status** Not processed

```
51 {
52     while (1)
53     {

54         if (rand ())

55     {
56         double_release_001_tsk_001 (NULL);
57     }
```

01.w_Defects/double_release.c:105

Level Medium**Status** Not processed

```
102 {  
103   while (1)  
104   {  
  
105     if (rand ())  
  
106     {  
107       double_release_002_tsk_001 (NULL);  
108     }  
}
```

01.w_Defects/double_release.c:151

Level Medium**Status** Not processed

```
148 {  
149   while (1)  
150   {  
  
151     if (rand ())  
  
152     {  
153       double_release_003_tsk_001 (NULL);  
154     }  
}
```

01.w_Defects/double_release.c:196

Level Medium**Status** Not processed

```
193 {  
194   while (1)  
195   {  
  
196     if (rand ())  
}
```

```
197     {  
198         double_release_004_tsk_001 (NULL);  
199     }
```

01.w_Defects/double_release.c:245

Level Medium

Status Not processed

```
242 {  
243     while (1)  
244     {  
  
245         if (rand ())  
  
246             {  
247                 double_release_005_tsk_001 (NULL);  
248             }  
}
```

01.w_Defects/double_release.c:281

Level Medium

Status Not processed

```
278 pthread_create (& tid1, NULL, double_release_006_tsk_001, NULL);  
279 pthread_join (tid1, NULL);  
280  
  
281 if(rand())  
  
282     pthread_mutex_unlock (double_release_006_glb_mutex);  
283     pthread_mutex_unlock (double_release_006_glb_mutex);/*Tool should detect this  
line as error*/ /*ERROR:Double UnLock*/  
284     pthread_mutex_destroy (double_release_006_glb_mutex);
```

01.w_Defects/double_release.c:292

Level Medium

Status Not processed

```
289 {  
290   while (1)  
291 {  
  
292     if (rand ())  
  
293     {  
294       double_release_006_tsk_001 (NULL);  
295     }  

```

01.w_Defects/free_null_pointer.c:392

Level Medium

Status Not processed

```
389  
390 free_null_pointer_011_u_001 * free_null_pointer_011_func_002 ()  
391 {  
  
392   int flag = rand();  
  
393   switch (flag)  
394   {  
395     case 1:  

```

01.w_Defects/func_pointer.c:330

Level Medium

Status Not processed

```
327  
328 func_pointer_009_u_001 * func_pointer_009_func_001 (void)  
329 {  
  
330   int flag = rand();  

```

```
331 flag = 1;
332 func_pointer_009_u_001 *u;
333 switch (flag)
```

01.w_Defects/insign_code.c:22

Level Medium

Status Not processed

```
19 int i;
20 int j;
21
22 i = rand();
23 j = i - 1;
24 i = j + 1; /*Tool should detect this line as error*/ /*ERROR:Useless Assignment */
25 }
```

01.w_Defects/memory_allocation_failure.c:150

Level Medium

Status Not processed

```
147 void memory_allocation_failure_005 ()
148 {
149     int ret;
150     ret = memory_allocation_failure_005_func_001 (rand());
151     if(ret >= 0)
152         if(vptr != NULL)
153             free(vptr);
```

01.w_Defects/memory_leak.c:197

Level Medium

Status Not processed

```
194 void memory_leak_007 ()  
195 {  
196     int ret;  
  
197     ret = memory_leak_007_func_001 (rand());  
  
198     if(ret == 0)  
199         if(vptr!=NULL)  
200     {
```

01.w_Defects/not_return.c:29

Level Medium

Status Not processed

```
26 void not_return_001 ()  
27 {  
28     int ret;  
  
29     ret = not_return_001_func_001(rand());  
  
30     sink = ret;  
31 }  
32
```

01.w_Defects/not_return.c:55

Level Medium

Status Not processed

```
52 void not_return_002 ()  
53 {  
54     int ret;  
  
55     ret = not_return_002_func_001(rand(), rand());  
  
56     sink = ret;  
57 }  
58
```

01.w_Defects/not_return.c:81

Level Medium**Status** Not processed

```
78 void not_return_003 ()  
79 {  
80     int ret;  
  
81     ret = not_return_003_func_001(rand());  
  
82     sink = ret;  
83 }  
84
```

01.w_Defects/not_return.c:104

Level Medium**Status** Not processed

```
101 void not_return_004 ()  
102 {  
103     int ret;  
  
104     ret = not_return_004_func_001(rand());  
  
105     sink = ret;  
106 }  
107
```

01.w_Defects/null_pointer.c:104

Level Medium**Status** Not processed

```
101 void null_pointer_006 ()  
102 {  
103     int *p;  
  
104     p = (int *)(intptr_t)rand();
```

```
105 *p = 1; /*Tool should detect this line as error*/ /*ERROR:NULL pointer dereference*/
106 }
107
```

01.w_Defects/overrun_st.c:181

Level Medium**Status** Not processed

```
178 {
179     int buf[5];
180     int index;

181     index = rand();

182     buf[index] = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
183     sink = buf[idx];
184 }
```

01.w_Defects/overrun_st.c:442

Level Medium**Status** Not processed

```
439     int *p;
440     int index;
441     p = buf;

442     index = rand();

443     *(p + index) = 1; /*Tool should detect this line as error*/ /*ERROR: buffer overrun */
444     sink = buf[idx];
445 }
```

01.w_Defects/redundant_cond.c:25

Level Medium

Status Not processed

```
22 int b = 0;  
23 int ret;  
24
```

25 a = rand();

```
26 if ((5 < a) && (10 < a)) /*Tool should detect this line as error*/ /*ERROR:Redundant  
condition*/  
27 {  
28     b += a;
```

01.w_Defects/redundant_cond.c:44

Level Medium

Status Not processed

```
41 int b = 0;  
42 int ret;  
43
```

44 a = rand();

```
45 if ((a < 5) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:Redundant  
condition*/  
46 {  
47     b += a;
```

01.w_Defects/redundant_cond.c:63

Level Medium

Status Not processed

```
60 int b = 0;  
61 int ret;  
62
```

63 a = rand();

```
64 if (((0 < a) && (a < 10)) && ((2 < a) && (a < 8))/*Tool should detect this line as error*/
/*ERROR:Redundant condition*/
65 {
66     b += a;
```

01.w_Defects/redundant_cond.c:82

Level Medium**Status** Not processed

```
79 int b = 0;
80 int ret;
81
```

82 a = rand();

```
83 if (((0 < a) && (a < 8)) && ((5 < a) && (a < 10))/*Tool should detect this line as error*/
/*ERROR:Redundant condition*/
84 {
85     b += a;
```

01.w_Defects/redundant_cond.c:101

Level Medium**Status** Not processed

```
98 int b = 0;
99 int ret;
100
```

101 a = rand();

```
102 if ((5 < a) || (10 < a))/*Tool should detect this line as error*/ /*ERROR:Redundant
condition*/
103 {
104     b += a;
```

01.w_Defects/redundant_cond.c:120

Level Medium**Status** Not processed

```
117 int b = 0;  
118 int ret;  
119
```

```
120 a = rand();
```

```
121 if (a < 5)  
122 {  
123   if (a < 10)/*Tool should detect this line as error*/ /*ERROR:Redundant condition*/
```

01.w_Defects/redundant_cond.c:142

Level Medium**Status** Not processed

```
139 int b;  
140 int ret;  
141
```

```
142 a = rand();
```

```
143 b = ((5 < a) && (10 < a)) ? 0 : 1;/*Tool should detect this line as error*/ /*ERROR:  
Redundant condition*/  
144 ret = b;  
145   sink = ret;
```

01.w_Defects/redundant_cond.c:176

Level Medium**Status** Not processed

```
173 int b = 0;  
174 int ret;  
175
```

176 a = rand();

177 while ((5 < a) && (10 < a))/*Tool should detect this line as error*/ /*ERROR:
Redundant condition*/
178 {
179 b += a;

01.w_Defects/redundant_cond.c:196

Level Medium

Status Not processed

193 int b = 0;
194 int ret;
195

196 a = rand();

197 while ((a < 5) && (a < 10))/*Tool should detect this line as error*/ /*ERROR:
Redundant condition*/
198 {
199 b += a;

01.w_Defects/redundant_cond.c:216

Level Medium

Status Not processed

213 int b = 0;
214 int ret;
215

216 a = rand();

217 while (((0 < a) && (a < 10)) && ((2 < a) && (a < 8)))/*Tool should detect this line as
error*/ /*ERROR:Redundant condition*/
218 {
219 b += a;

01.w_Defects/redundant_cond.c:236

Level Medium**Status** Not processed

```
233 int b = 0;  
234 int ret;  
235
```

236 a = rand();

```
237 while (((0 < a) && (a < 8)) && ((5 < a) && (a < 10))/*Tool should detect this line as  
error*/ /*ERROR:Redundant condition*/  
238 {  
239   b += a;
```

01.w_Defects/redundant_cond.c:256

Level Medium**Status** Not processed

```
253 int b = 0;  
254 int ret;  
255
```

256 a = rand();

```
257 while ((5 < a) || (10 < a))/*Tool should detect this line as error*/ /*ERROR:Redundant  
condition*/  
258 {  
259   b += a;
```

01.w_Defects/redundant_cond.c:276

Level Medium**Status** Not processed

```
273 int b = 0;  
274 int ret;  
275
```

```
276 a = rand();
```

```
277 do  
278 {  
279   b += a;
```

01.w_Defects/sign_conv.c:165

Level Medium

Status Not processed

```
162  
163 /*      0 rand() 2147483647 RAND_MAX */  
164 /* 1073741823 rand() 1073741823 1073741824 */
```

```
165 a = rand() - 1073741823;
```

```
166  
167 ret = a; /*Tool should detect this line as error*/ /*Integer sign lost because of unsigned  
cast */  
168     sink = ret;
```

01.w_Defects/sleep_lock.c:70

Level Medium

Status Not processed

```
67 {  
68   while (1)  
69   {  
  
70     if (rand())  
  
71     {  
72       sleep_lock_001_tsk_001(NULL);  
73     }
```

01.w_Defects/sleep_lock.c:148

Level Medium**Status** Not processed

```
145 {  
146   while (1)  
147   {  
  
148     if (rand())  
  
149     {  
150       sleep_lock_002_tsk_001(NULL);  
151     }  
}
```

01.w_Defects/sleep_lock.c:202

Level Medium**Status** Not processed

```
199 {  
200   while (1)  
201   {  
  
202     if (rand())  
  
203     {  
204       sleep_lock_003_tsk_001(NULL);  
205     }  
}
```

01.w_Defects/uninit_memory_access.c:385

Level Medium**Status** Not processed

```
382  
383 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001 ()  
384 {  
  
385   int flag = rand();
```

```
386 uninits_memory_access_014_u_001 *u;
387 switch (flag)
388 {
```

01.w_Defects/zero_division.c:153

Level Medium

Status Not processed

```
150 int dividend = 1000;
151 int divisor;
152 int ret;

153 divisor = rand();
```

```
154 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:division by
zero */
155 }
156
```

02.wo_Defects/bit_shift.c:120

Level Medium

Status Not processed

```
117 int a = 1;
118 int shift;
119 int ret;

120 shift = rand() % 32;
```

```
121 ret = a << shift; /*Tool should not detect this line as error*/ /*NO ERROR:Bit shift
error*/
122     sink = ret;
123 }
```

02.wo_Defects/conflicting_cond.c:24

Level Medium**Status** Not processed

```
21 int b = 0;  
22 int ret;  
23
```

```
24 a = rand();
```

```
25 if ((a == 0) || (a == 1)) /*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/
```

```
26 {  
27     b += a;
```

02.wo_Defects/conflicting_cond.c:43

Level Medium**Status** Not processed

```
40 int b = 0;  
41 int ret;  
42
```

```
43 a = rand();
```

```
44 if (!(a < 5) || (10 < a)) /*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/
```

```
45 {  
46     b += a;
```

02.wo_Defects/conflicting_cond.c:62

Level Medium**Status** Not processed

```
59 int b = 0;  
60 int ret;  
61
```

```
62 a = rand();
```

```
63 if (((0 < a) && (a < 2)) || ((8 < a) && (a < 10)))/*Tool should not detect this line as  
error*/ /*No ERROR:contradict condition*/  
64 {  
65     b += a;
```

02.wo_Defects/conflicting_cond.c:81

Level Medium

Status Not processed

```
78 int b = 0;  
79 int ret;  
80
```

```
81 a = rand();
```

```
82 if (a < 5)  
83 {  
84     a += 10;
```

02.wo_Defects/conflicting_cond.c:104

Level Medium

Status Not processed

```
101 int b;  
102 int ret;  
103
```

```
104 a = rand();
```

```
105 b = ((a == 0) || (a == 1)) ? 0 : 1;/*Tool should not detect this line as error*/ /*No  
ERROR:contradict condition*/  
106 ret = b;  
107     sink = ret;
```

02.wo_Defects/conflicting_cond.c:138

Level Medium**Status** Not processed

```
135 int b = 0;  
136 int ret;  
137
```

```
138 a = rand();
```

```
139 while ((a == 0) || (a == 1))/*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/
```

```
140 {  
141   b += a;
```

02.wo_Defects/conflicting_cond.c:158

Level Medium**Status** Not processed

```
155 int b = 0;  
156 int ret;  
157
```

```
158 a = rand();
```

```
159 while (! ((a < 5) || (10 < a)))/*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/
```

```
160 {  
161   b += a;
```

02.wo_Defects/conflicting_cond.c:178

Level Medium**Status** Not processed

```
175 int b = 0;  
176 int ret;  
177
```

```
178 a = rand();
```

```
179 while (((0 < a) && (a < 2)) || ((8 < a) && (a < 10))/*Tool should not detect this line as  
error*/ /*No ERROR:contradict condition*/  
180 {  
181   b += a;
```

02.wo_Defects/conflicting_cond.c:199

Level Medium

Status Not processed

```
196  
197 do  
198 {
```

```
199 a = rand();
```

```
200 }  
201 while ((a == 0) || (a == 1));/*Tool should not detect this line as error*/ /*No ERROR:  
contradict condition*/  
202 ret = a;
```

02.wo_Defects/data_lost.c:160

Level Medium

Status Not processed

```
157 {  
158   short ret;  
159   int a;
```

```
160   a = rand() % 0x8000;
```

```
161   ret = a; /*Tool should not detect this line as error*/ /*No ERROR:Integer precision  
lost because of cast*/  
162     sink = ret;  
163 }
```

02.wo_Defects/data_overflow.c:205

Level Medium**Status** Not processed

```
202 int max = 0x7fffffff;  
203 int d;  
204 int ret;
```

```
205 d = rand() % 2;
```

```
206 ret = max + d; /*Tool should not detect this line as error*/ /*No ERROR:Data  
Overflow*/  
207     sink = ret;  
208 }
```

02.wo_Defects/double_release.c:53

Level Medium**Status** Not processed

```
50 {  
51   while (1)  
52 {
```

```
53     if (rand ())
```

```
54     {  
55       double_release_001_tsk_001 (NULL);  
56     }
```

02.wo_Defects/double_release.c:106

Level Medium**Status** Not processed

```
103 {  
104     while (1)  
105     {  
  
106         if (rand ())  
  
107         {  
108             double_release_002_tsk_001 (NULL);  
109         }  
110     }  
111 }
```

02.wo_Defects/double_release.c:154

Level Medium

Status Not processed

```
151 {  
152     while (1)  
153     {  
  
154         if (rand ())  
  
155         {  
156             double_release_003_tsk_001 (NULL);  
157         }  
158     }  
159 }
```

02.wo_Defects/double_release.c:201

Level Medium

Status Not processed

```
198 {  
199     while (1)  
200     {  
  
201         if (rand ())  
  
202         {  
203             double_release_004_tsk_001 (NULL);  
204         }  
205     }  
206 }
```

02.wo_Defects/double_release.c:250

Level Medium**Status** Not processed

```
247 {  
248   while (1)  
249   {  
  
250       if (rand ())  
  
251       {  
252           double_release_005_tsk_001 (NULL);  
253       }  
}
```

02.wo_Defects/double_release.c:292

Level Medium**Status** Not processed

```
289 {  
290   while (1)  
291   {  
  
292       if (rand ())  
  
293       {  
294           double_release_006_tsk_001 (NULL);  
295       }  
}
```

02.wo_Defects/free_null_pointer.c:364

Level Medium**Status** Not processed

```
361 static free_null_pointer_011_u_001 *u;  
362 free_null_pointer_011_u_001 * free_null_pointer_011_func_001 ()  
363 {  
  
364     int flag = rand();  
}
```

```
365 flag = 1;
366 switch (flag)
367 {
```

02.wo_Defects/func_pointer.c:341

Level Medium

Status Not processed

```
338
339 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
340 {

341 int flag = rand();

342 flag = 1;
343 func_pointer_009_u_001 *u;
344 switch (flag)
```

02.wo_Defects/insign_code.c:23

Level Medium

Status Not processed

```
20 int i;
21 int j;
22

23 i = rand();

24 j = i - 1;
25 i = j - 1; /*Tool should not detect this line as error*/ /*No ERROR:Useless Assignment*/
26 printf("%d",i);
```

02.wo_Defects/memory_leak.c:200

Level Medium

Status Not processed

```
197 void memory_leak_007 ()  
198 {  
199     int ret;  
  
200     ret = memory_leak_007_func_001 (rand());  
  
201     if(ret >= 0 )  
202         if(vptr!=NULL)  
203             {
```

02.wo_Defects/not_return.c:34

Level Medium

Status Not processed

```
31 void not_return_001 ()  
32 {  
33     int ret;  
  
34     ret = not_return_001_func_001(rand());  
  
35     sink = ret;  
36 }  
37
```

02.wo_Defects/not_return.c:61

Level Medium

Status Not processed

```
58 void not_return_002 ()  
59 {  
60     int ret;  
  
61     ret = not_return_002_func_001(rand(), rand());
```

```
62     sink = ret;
63 }
64
```

02.wo_Defects/not_return.c:87

Level Medium

Status Not processed

```
84 void not_return_003 ()
85 {
86     int ret;

87     ret = not_return_003_func_001(rand());

88     sink = ret;
89 }
90
```

02.wo_Defects/not_return.c:111

Level Medium

Status Not processed

```
108 void not_return_004 ()
109 {
110     int ret;

111     ret = not_return_004_func_001(rand());

112     sink = ret;
113 }
114
```

02.wo_Defects/overrun_st.c:183

Level Medium

Status Not processed

```
180 {  
181     int buf[5];  
182     int index;  
  
183     index = rand() % 5;  
  
184     buf[index] = 1; /*Tool should not detect this line as error*/ /*No ERROR: buffer  
overrun */  
185     sink = buf[idx];  
186 }
```

02.wo_Defects/overrun_st.c:443

Level Medium

Status Not processed

```
440     int *p;  
441     int index;  
442     p = buf;  
  
443     index = rand() % 5;
```

```
444     *(p + index) = 1; /*Tool should not detect this line as error*/ /*No ERROR: buffer  
overrun */  
445 }  
446
```

02.wo_Defects/redundant_cond.c:26

Level Medium

Status Not processed

```
23 int b = 0;  
24 int ret;  
25
```

```
26 a = rand();
```

```
27 if ( a < 10 ) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/
```

```
28 {  
29   b += a;
```

02.wo_Defects/redundant_cond.c:45

Level Medium

Status Not processed

```
42 int b = 0;  
43 int ret;  
44
```

45 a = rand();

46 if (a < 5) /*Tool should not detect this line as error*/ /*No ERROR:Redundant condition*/

```
47 {  
48   b += a;
```

02.wo_Defects/redundant_cond.c:64

Level Medium

Status Not processed

```
61 int b = 0;  
62 int ret;  
63
```

64 a = rand();

65 if (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:Redundant condition*/

```
66 {  
67   b += a;
```

02.wo_Defects/redundant_cond.c:83

Level Medium

Status Not processed

```
80 int b = 0;  
81 int ret;  
82  
  
83 a = rand();  
  
84 if ( a < 10 )/*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
85 {  
86   b += a;
```

02.wo_Defects/redundant_cond.c:102

Level Medium

Status Not processed

```
99 int b = 0;  
100 int ret;  
101
```

```
102 a = rand();
```

```
103 if (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
104 {  
105   b += a;
```

02.wo_Defects/redundant_cond.c:121

Level Medium

Status Not processed

```
118 int b = 0;  
119 int ret;  
120
```

```
121 a = rand();
```

```
122 if (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
123 {  
124   b += a;
```

02.wo_Defects/redundant_cond.c:141

Level Medium

Status Not processed

```
138 int b;  
139 int ret;  
140
```

141 a = rand();

```
142 b = ((a < 10) ) ? 0 : 1; /*Tool should not detect this line as error*/ /*No ERROR:  
Redundant condition*/  
143 ret = b;  
144   sink = ret;
```

02.wo_Defects/redundant_cond.c:175

Level Medium

Status Not processed

```
172 int b = 0;  
173 int ret;  
174
```

175 a = rand();

```
176 while (a < 10) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
177 {  
178   b += a;
```

02.wo_Defects/redundant_cond.c:195

Level Medium

Status Not processed

```
192 int b = 0;  
193 int ret;  
194
```

195 a = rand();

```
196 while ((a < 5)) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
197 {  
198   b += a;
```

02.wo_Defects/redundant_cond.c:215

Level Medium

Status Not processed

```
212 int b = 0;  
213 int ret;  
214
```

215 a = rand();

```
216 while ((a < 8)) /*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
217 {  
218   b += a;
```

02.wo_Defects/redundant_cond.c:235

Level Medium

Status Not processed

```
232 int b = 0;  
233 int ret;  
234
```

235 a = rand();

```
236 while (a < 10)/*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
237 {  
238   b += a;
```

02.wo_Defects/redundant_cond.c:255

Level Medium

Status Not processed

```
252 int b = 0;  
253 int ret;  
254
```

255 a = rand();

```
256 while ((a < 10))/*Tool should not detect this line as error*/ /*No ERROR:Redundant  
condition*/  
257 {  
258   b += a;
```

02.wo_Defects/redundant_cond.c:275

Level Medium

Status Not processed

```
272 int b = 0;  
273 int ret;  
274
```

275 a = rand();

```
276 do {  
277   // JDR: cast to unsigned to avoid UB  
278   b += (unsigned)a;
```

02.wo_Defects/sign_conv.c:165

Level Medium**Status** Not processed

```
162 unsigned int ret;  
163  
164 /* 0 <= rand() <= 2147483647 (RAND_MAX) */
```

```
165 a = rand();
```

```
166  
167 ret = a; /*Tool should not detect this line as error*/ /*NO ERROR : Integer sign lost  
because of unsigned cast */  
168     sink = ret;
```

02.wo_Defects/uninit_memory_access.c:401

Level Medium**Status** Not processed

```
398  
399 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001 ()  
400 {
```

```
401 int flag = rand();
```

```
402 flag = 1;  
403 uninit_memory_access_014_u_001 *u;  
404 switch (flag)
```

02.wo_Defects/zero_division.c:151

Level Medium**Status** Not processed

```
148 int dividend = 1000;  
149 int divisor;  
150 int ret;
```

```
151 divisor = rand();  
  
152 if (divisor != 0)  
153 {  
154     ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */
```

Block critical section (C/C++)

Description

Calls to blocking functions (for example: sleep, getc, fgets, read, recv etc.) inside a critical section can lead to bad performance or even deadlock.

Example

The following example shows a code snippet that contains calls to blocking functions:

```
void testBlockInCriticalSectionWithStdMutex() {  
    std::mutex m;  
    m.lock();  
    sleep(3);  
    getc();  
    fgets();  
    read();  
    recv();  
    m.unlock();  
}
```

Recommendations

- If possible, avoid using blocking functions inside a critical section in the release version of the code.

Links

1. Critical Sections, Locking, and Deadlocks

2. CWE-667: Improper Locking

Vulnerability Entries

01.w_Defects/sleep_lock.c:50

Level Low**Status** Not processed

```
47     pthread_mutex_lock(sleep_lock_001_glb_mutex);
48     sleep_lock_001_glb_data = (sleep_lock_001_glb_data % 100) + 1;
49 #ifndef __NO_COVERITY__
50     sleep(3600);/*Tool should detect this line as error*/ /*Error:Long Sleep */
51 #endif
52     pthread_mutex_unlock(sleep_lock_001_glb_mutex);
53     return NULL;
```

Trace

rand()

01.w_Defects/sleep_lock.c:70

```
67 {
68     while (1)
69     {
70         if (rand())
71         {
72             sleep_lock_001_tsk_001(NULL);
73         }
    }
```

```
sleep(3600)
```

01.w_Defects/sleep_lock.c:50

```
47 pthread_mutex_lock(sleep_lock_001_glb_mutex);
48 sleep_lock_001_glb_data = (sleep_lock_001_glb_data % 100) + 1;
49 #ifndef __NO_COVERITY__
50 sleep(3600);/*Tool should detect this line as error*/ /*Error:Long Sleep */
51 #endif
52 pthread_mutex_unlock(sleep_lock_001_glb_mutex);
53 return NULL;
```

01.w_Defects/sleep_lock.c:123

Level Low

Status Not processed

```
120 #ifndef __NO_COVERITY__
121
122 sleep_lock_002_glb_size =
123         recv(sock, sleep_lock_002_glb_data, 256, 0);
124 sleep(3600);/*Tool should detect this line as error*/ /*Error:Long Sleep */
125 #endif
126
```

Trace

```
rand()
```

01.w_Defects/sleep_lock.c:148

```
145 {
146     while (1)
147     {
148         if (rand())
```

```
149     {  
150         sleep_lock_002_tsk_001(NULL);  
151     }
```

```
recv(sock, sleep_lock_002_glb_data, 256,  
0)
```

01.w_Defects/sleep_lock.c:123

```
120 #ifndef __NO_COVERITY__  
121  
122 sleep_lock_002_glb_size =  
  
123     recv(sock, sleep_lock_002_glb_data, 256, 0);  
  
124 sleep(3600);/*Tool should detect this line as error*/ /*Error:Long Sleep */  
125 #endif  
126
```

01.w_Defects/sleep_lock.c:124

Level Low

Status Not processed

```
121  
122 sleep_lock_002_glb_size =  
123     recv(sock, sleep_lock_002_glb_data, 256, 0);  
  
124 sleep(3600);/*Tool should detect this line as error*/ /*Error:Long Sleep */  
  
125 #endif  
126  
127 /* Lock or unlock */
```

Trace

rand()

01.w_Defects/sleep_lock.c:148

```
145 {  
146     while (1)  
147     {  
  
148         if (rand())  
  
149         {  
150             sleep_lock_002_tsk_001(NULL);  
151         }  
}
```

sleep(3600)

01.w_Defects/sleep_lock.c:124

```
121  
122 sleep_lock_002_glb_size =  
123     recv(sock, sleep_lock_002_glb_data, 256, 0);  
  
124 sleep(3600);/*Tool should detect this line as error*/ /*Error:Long Sleep */  
  
125 #endif  
126  
127 /* Lock or unlock */
```

01.w_Defects/sleep_lock.c:173

Level Low**Status** Not processed

```
170 {  
171 #ifndef __NO_COVERITY__  
172
```

```
173 sleep(3600);/*Tool should detect this line as error*/ /*Error:Long Sleep */
```

```
174 #endif
```

```
175 }  
176
```

Trace

```
rand()
```

```
01.w_Defects/sleep_lock.c:202
```

```
199 {  
200   while (1)  
201   {  
  
202     if (rand())  
  
203     {  
204       sleep_lock_003_tsk_001(NULL);  
205     }
```

```
sleep(3600)
```

```
01.w_Defects/sleep_lock.c:173
```

```
170 {  
171 #ifndef __NO_COVERITY__  
172  
  
173 sleep(3600);/*Tool should detect this line as error*/ /*Error:Long Sleep */  
  
174 #endif  
175 }  
176
```

Dead store (C/C++)

Description

A local variable is assigned a value but is not read by any subsequent instruction. Dead stores waste processor time and memory and may indicate significant logic errors.

Example

In the following example, a variable is created but will never be used:

```
UIPasteboard *pb1 = [UIPasteboard generalPasteboard];
```

The correct version:

```
UIPasteboard *pb1 = [UIPasteboard generalPasteboard];
pb1.persistent = YES;
```

Recommendations

- Remove the unused variable from the source.
- In compiler theory, there are DCE (dead code elimination) algorithms that remove dead code.

Links

1. CWE-563: Assignment to Variable without Use ('Unused Variable')
2. CWE-561: Dead Code

Vulnerability Entries

01.w_Defects/buffer_underrun_dynamic.c:701

Level Low

Status Not processed

```
698 {
699     memcpy (newTest,test,10);
700     char c ;
```

```
701     c = newTest[-10]; /*Tool should detect this line as error*/ /*ERROR:Buffer
Underrun*/
```

```
702     free(newTest);
703 }
704 }
```

Trace

newTest[-10]

01.w_Defects/buffer_underrun_dynamic.c:701

```
698 {  
699     memcpy (newTest,test,10);  
700     char c ;  
  
701     c = newTest[-10]; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
702     free(newTest);  
703 }  
704 }
```

newTest[-10]

01.w_Defects/buffer_underrun_dynamic.c:701

```
698 {  
699     memcpy (newTest,test,10);  
700     char c ;  
  
701     c = newTest[-10]; /*Tool should detect this line as error*/ /*ERROR:Buffer  
Underrun*/  
  
702     free(newTest);  
703 }  
704 }
```

01.w_Defects/free_nondynamically_allocated_memory.c:98

Level Low**Status** Not processed

```
95 {  
96     int b=2;float c=3.5; double d=4.5;  
97     char* ptr1="a";  
  
98     int* ptr2=&b;
```

```
99 float* ptr3=&c;
100 double* ptr4=&d;
101
```

Trace

&b

01.w_Defects/free_nondynamically_allocated_memory.c:98

```
95 {
96     int b=2;float c=3.5; double d=4.5;
97     char* ptr1="a";

98     int* ptr2=&b;

99     float* ptr3=&c;
100    double* ptr4=&d;
101
```

&b

01.w_Defects/free_nondynamically_allocated_memory.c:98

```
95 {
96     int b=2;float c=3.5; double d=4.5;
97     char* ptr1="a";

98     int* ptr2=&b;

99     float* ptr3=&c;
100    double* ptr4=&d;
101
```

01.w_Defects/free_nondynamically_allocated_memory.c:99

Level Low

Status Not processed

```
96 int b=2;float c=3.5; double d=4.5;  
97 char* ptr1="a";  
98 int* ptr2=&b;
```

```
99 float* ptr3=&c;
```

```
100 double* ptr4=&d;  
101  
102
```

Trace

```
&c
```

```
01.w_Defects/free_nondynamically_allocated_memory.c:99
```

```
96 int b=2;float c=3.5; double d=4.5;  
97 char* ptr1="a";  
98 int* ptr2=&b;
```

```
99 float* ptr3=&c;
```

```
100 double* ptr4=&d;  
101  
102
```

```
&c
```

```
01.w_Defects/free_nondynamically_allocated_memory.c:99
```

```
96 int b=2;float c=3.5; double d=4.5;  
97 char* ptr1="a";  
98 int* ptr2=&b;
```

```
99 float* ptr3=&c;
```

```
100 double* ptr4=&d;  
101  
102
```

01.w_Defects/free_null_pointer.c:419

Level Low**Status** Not processed

```
416 int ret;
417 free_null_pointer_011_u_001 *p;
418 p = free_null_pointer_011_func_001 ();

419 ret = p->b;

420 p = free_null_pointer_011_func_002 ();
421 }
422
```

Trace

p->b

01.w_Defects/free_null_pointer.c:419

```
416 int ret;
417 free_null_pointer_011_u_001 *p;
418 p = free_null_pointer_011_func_001 ();

419 ret = p->b;

420 p = free_null_pointer_011_func_002 ();
421 }
422
```

p->b

01.w_Defects/free_null_pointer.c:419

```
416 int ret;
417 free_null_pointer_011_u_001 *p;
418 p = free_null_pointer_011_func_001 ();

419 ret = p->b;

420 p = free_null_pointer_011_func_002 ();
```

```
421 }
422
```

01.w_Defects/free_null_pointer.c:420

Level Low

Status Not processed

```
417 free_null_pointer_011_u_001 *p;
418 p = free_null_pointer_011_func_001 ();
419 ret = p->b;
```

```
420 p = free_null_pointer_011_func_002 ();
```

```
421 }
422
423 /*
```

Trace

free_null_pointer_011_func_002 ()

01.w_Defects/free_null_pointer.c:420

```
417 free_null_pointer_011_u_001 *p;
418 p = free_null_pointer_011_func_001 ();
419 ret = p->b;
```

```
420 p = free_null_pointer_011_func_002 ();
```

```
421 }
422
423 /*
```

free_null_pointer_011_func_002 ()

01.w_Defects/free_null_pointer.c:420

```
417 free_null_pointer_011_u_001 *p;
418 p = free_null_pointer_011_func_001 ();
419 ret = p->b;

420 p = free_null_pointer_011_func_002 ();

421 }
422
423 /*
```

01.w_Defects/free_null_pointer.c:452

Level Low**Status** Not processed

```
449 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
450 {
451   if(flag == 10)

452   a = *(ptr+1);

453 }
454
455 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
```

Trace

*(ptr+1)

01.w_Defects/free_null_pointer.c:452

```
449 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
450 {
451   if(flag == 10)

452   a = *(ptr+1);
```

```
453 }  
454  
455 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
```

```
*(ptr+1)
```

01.w_Defects/free_null_pointer.c:452

```
449 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)  
450 {  
451   if(flag == 10)  
  
452   a = *(ptr+1);  
  
453 }  
454  
455 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
```

01.w_Defects/function_return_value_unchecked.c:393

Level Low

Status Not processed

```
390 case 6:  
391   puts("TEST"); /*Tool should detect this line as error*/ /*ERROR:Return value of  
function never checked*/  
392 {  
  
393     i=20;  
  
394   }  
395   break;  
396 default:
```

Trace

i=20

01.w_Defects/function_return_value_unchecked.c:393

```
390 case 6:  
391     puts("TEST"); /*Tool should detect this line as error*/ /*ERROR:Return  
value of function never checked*/  
392     {  
  
393         i=20;  
  
394     }  
395     break;  
396 default:
```

i=20

01.w_Defects/function_return_value_unchecked.c:393

```
390 case 6:  
391     puts("TEST"); /*Tool should detect this line as error*/ /*ERROR:Return  
value of function never checked*/  
392     {  
  
393         i=20;  
  
394     }  
395     break;  
396 default:
```

01.w_Defects/function_return_value_unchecked.c:398

Level Low**Status** Not processed

```
395     break;  
396 default:  
397     {
```

```
398         i=10;
```

```
399     break;
400 }
401 }
```

Trace

i=10

01.w_Defects/function_return_value_unchecked.c:398

```
395     break;
396 default:
397 {

398     i=10;

399     break;
400 }
401 }
```

i=10

01.w_Defects/function_return_value_unchecked.c:398

```
395     break;
396 default:
397 {

398     i=10;

399     break;
400 }
401 }
```

01.w_Defects/func_pointer.c:34

Level Low

Status Not processed

```
31 void func_pointer_001_func_001 ()  
32 {  
33     int a ;  
  
34     a =10;  
  
35 }  
36  
37 void func_pointer_001 ()
```

Trace

a =10

01.w_Defects/func_pointer.c:34

```
31 void func_pointer_001_func_001 ()  
32 {  
33     int a ;  
  
34     a =10;  
  
35 }  
36  
37 void func_pointer_001 ()
```

a =10

01.w_Defects/func_pointer.c:34

```
31 void func_pointer_001_func_001 ()  
32 {  
33     int a ;  
  
34     a =10;  
  
35 }  
36  
37 void func_pointer_001 ()
```

01.w_Defects/func_pointer.c:42

Level Low**Status** Not processed

```
39 int (*func)();  
40 int ret;  
41 func = (int (*)())func_pointer_001_func_001;  
  
42 ret = func();/*Tool should detect this line as error*/ /*ERROR:Bad function pointer  
casting*/  
  
43 }  
44  
45 /*
```

Trace

func()

01.w_Defects/func_pointer.c:42

```
39 int (*func)();  
40 int ret;  
41 func = (int (*)())func_pointer_001_func_001;  
  
42 ret = func();/*Tool should detect this line as error*/ /*ERROR:Bad function  
pointer casting*/  
  
43 }  
44  
45 /*
```

func()

01.w_Defects/func_pointer.c:42

```
39 int (*func)();  
40 int ret;  
41 func = (int (*)())func_pointer_001_func_001;  
  
42 ret = func();/*Tool should detect this line as error*/ /*ERROR:Bad function
```

pointer casting*/

```
43 }  
44  
45 /*
```

01.w_Defects/func_pointer.c:81

Level Low

Status Not processed

```
78 {  
79     float (*func)(long , int);  
80     func = (float (*)(long , int ))func_pointer_003_func_001;  
  
81         ret = func(1, 2);/*Tool should detect this line as error*/ /*ERROR:Bad  
function pointer casting*/  
  
82 }  
83 }  
84
```

Trace

func(1, 2)

01.w_Defects/func_pointer.c:81

```
78 {  
79     float (*func)(long , int);  
80     func = (float (*)(long , int ))func_pointer_003_func_001;  
  
81         ret = func(1, 2);/*Tool should detect this line as error*/ /*ERROR:Bad  
function pointer casting*/  
  
82 }  
83 }  
84
```

```
func(1, 2)
```

01.w_Defects/func_pointer.c:81

```
78  {
79      float (*func)(long , int);
80      func = (float (*)(long , int ))func_pointer_003_func_001;
81      ret = func(1, 2);/*Tool should detect this line as error*/ /*ERROR:Bad
function pointer casting*/
82  }
83 }
84
```

01.w_Defects/func_pointer.c:123

Level Low

Status Not processed

```
120      char str;
121      int (*fptr)(char *);
122      fptr = (int (*)( char *))func_pointer_004_func_001;
123      str = fptr(buf[j]);/*Tool should detect this line as error*/ /*ERROR:Bad function
pointer casting*/
124  }
125 }
126 }
```

Trace

```
fptr(buf[j])
```

01.w_Defects/func_pointer.c:123

```
120      char str;
121      int (*fptr)(char *);
122      fptr = (int (*)( char *))func_pointer_004_func_001;
```

```
123     str = fptra(buf[j]);/*Tool should detect this line as error*/ /*ERROR:Bad  
function pointer casting*/  
  
124     }  
125 }  
126 }
```

fptra(buf[j])

01.w_Defects/func_pointer.c:123

```
120     char str;  
121     int (*fptra)(char *);  
122     fptra = (int (*)( char *))func_pointer_004_func_001;  
  
123     str = fptra(buf[j]);/*Tool should detect this line as error*/ /*ERROR:Bad  
function pointer casting*/  
  
124     }  
125 }  
126 }
```

01.w_Defects/func_pointer.c:256

Level Low

Status Not processed

```
253 char **(*fptra)();  
254 char **doubleptr;  
255 fptra = (char ** (*) (void))func_pointer_006_func_002;  
  
256 doubleptr = fptra();/*Tool should detect this line as error*/ /*ERROR:Bad function  
pointer casting*/  
  
257 doubleptr = (char **)func_pointer_006_func_003();  
258 for(i=0;i<10;i++)  
259 {
```

Trace

fptr()

01.w_Defects/func_pointer.c:256

```
253 char **(*fptr)();  
254 char **doubleptr;  
255 fptr = (char ** (*) (void)) func_pointer_006_func_002;
```

256 doubleptr = fptr(); /*Tool should detect this line as error*/ /*ERROR:Bad
function pointer casting*/

```
257 doubleptr = (char **) func_pointer_006_func_003();  
258 for(i=0;i<10;i++)  
259 {
```

fptr()

01.w_Defects/func_pointer.c:256

```
253 char **(*fptr)();  
254 char **doubleptr;  
255 fptr = (char ** (*) (void)) func_pointer_006_func_002;
```

256 doubleptr = fptr(); /*Tool should detect this line as error*/ /*ERROR:Bad
function pointer casting*/

```
257 doubleptr = (char **) func_pointer_006_func_003();  
258 for(i=0;i<10;i++)  
259 {
```

01.w_Defects/func_pointer.c:265

Level Low**Status** Not processed

```
262             doubleptr[i][j] += 1;  
263         }  
264     }
```

```
265     doubleptr = (char **) func_pointer_006_func_004();
```

```
266 }  
267 }  
268
```

Trace

```
(char **)func_pointer_006_func_004()
```

01.w_Defects/func_pointer.c:265

```
262         doubleptr[i][j] += 1;  
263     }  
264 }  
  
265 doubleptr = (char **)func_pointer_006_func_004();  
  
266 }  
267 }  
268
```

```
(char **)func_pointer_006_func_004()
```

01.w_Defects/func_pointer.c:265

```
262         doubleptr[i][j] += 1;  
263     }  
264 }  
  
265 doubleptr = (char **)func_pointer_006_func_004();  
  
266 }  
267 }  
268
```

01.w_Defects/func_pointer.c:285

Level Low

Status Not processed

```
282 char ** (*fptr)(char []);  
283 char ** a = NULL;  
284 fptr = (char ** (*)(char []))func_pointer_007_func_001;  
  
285 a =fptr(buf);/*Tool should detect this line as error*/ /*ERROR:Bad function pointer  
casting*/  
  
286 }  
287  
288 /*
```

Trace

fptr(buf)

01.w_Defects/func_pointer.c:285

```
282 char ** (*fptr)(char []);  
283 char ** a = NULL;  
284 fptr = (char ** (*)(char []))func_pointer_007_func_001;  
  
285 a =fptr(buf);/*Tool should detect this line as error*/ /*ERROR:Bad function  
pointer casting*/  
  
286 }  
287  
288 /*
```

fptr(buf)

01.w_Defects/func_pointer.c:285

```
282 char ** (*fptr)(char []);  
283 char ** a = NULL;  
284 fptr = (char ** (*)(char []))func_pointer_007_func_001;  
  
285 a =fptr(buf);/*Tool should detect this line as error*/ /*ERROR:Bad function  
pointer casting*/  
  
286 }  
287  
288 /*
```

01.w_Defects/func_pointer.c:330

Level Low**Status** Not processed

```
327
328 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
329 {

330 int flag = rand();

331 flag = 1;
332 func_pointer_009_u_001 *u;
333 switch (flag)
```

Trace

rand()

01.w_Defects/func_pointer.c:330

```
327
328 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
329 {

330 int flag = rand();

331 flag = 1;
332 func_pointer_009_u_001 *u;
333 switch (flag)
```

rand()

01.w_Defects/func_pointer.c:330

```
327
328 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
329 {

330 int flag = rand();

331 flag = 1;
```

```
332 func_pointer_009_u_001 *u;
333 switch (flag)
```

01.w_Defects/func_pointer.c:353

Level Low**Status** Not processed

```
350 func_pointer_009_u_001 (*fptr)();
351 fptr = (func_pointer_009_u_001 (*)(void))func_pointer_009_func_001;
352 *p = fptr();/*Tool should detect this line as error*/ /*ERROR:Bad function pointer
casting*/
353 ret = p->b;
354 free(p);
355 p= NULL;
356 }
```

Trace

p->b

01.w_Defects/func_pointer.c:353

```
350 func_pointer_009_u_001 (*fptr)();
351 fptr = (func_pointer_009_u_001 (*)(void))func_pointer_009_func_001;
352 *p = fptr();/*Tool should detect this line as error*/ /*ERROR:Bad function pointer
casting*/
353 ret = p->b;
354 free(p);
355 p= NULL;
356 }
```

p->b

01.w_Defects/func_pointer.c:353

```
350 func_pointer_009_u_001 (*fptr)();
351 fptr = (func_pointer_009_u_001 (*)(void))func_pointer_009_func_001;
352 *p = fptr();/*Tool should detect this line as error*/ /*ERROR:Bad function
pointer casting*/
353 ret = p->b;
354 free(p);
355 p= NULL;
356 }
```

01.w_Defects/func_pointer.c:365

Level Low**Status** Not processed

```
362 void func_pointer_010_func_001 ()
363 {
364 int a;
365 a= 10;
366 }
367
368 void func_pointer_010 ()
```

Trace

a= 10

01.w_Defects/func_pointer.c:365

```
362 void func_pointer_010_func_001 ()
363 {
364 int a;
```

```
365 a= 10;  
  
366 }  
367  
368 void func_pointer_010 ()
```

a= 10

01.w_Defects/func_pointer.c:365

```
362 void func_pointer_010_func_001 ()  
363 {  
364     int a;  
  
365     a= 10;  
  
366 }  
367  
368 void func_pointer_010 ()
```

01.w_Defects/func_pointer.c:375

Level Low

Status Not processed

```
372     int ret;  
373     func = (int (*)(void))func_pointer_004_func_001;  
374     func1 = func;  
  
375     ret = func1();/*Tool should detect this line as error*/ /*ERROR:Bad function pointer  
casting*/
```

```
376 }  
377  
378 /*
```

Trace

func1()

01.w_Defects/func_pointer.c:375

```
372 int ret;
373 func = (int (*)(void))func_pointer_004_func_001;
374 func1 = func;
```

375 ret = func1();/*Tool should detect this line as error*/ /*ERROR:Bad function pointer casting*/

```
376 }
377
378 /*
```

func1()

01.w_Defects/func_pointer.c:375

```
372 int ret;
373 func = (int (*)(void))func_pointer_004_func_001;
374 func1 = func;
```

375 ret = func1();/*Tool should detect this line as error*/ /*ERROR:Bad function pointer casting*/

```
376 }
377
378 /*
```

01.w_Defects/func_pointer.c:498

Level Low**Status** Not processed

```
495 int flag;
496 int (*func_gbl)(int);
497 func_gbl = (int (*)(int))func_pointer_013_func_002;
```

498 flag = func_gbl(1);

```
499 }  
500  
501
```

Trace

```
func_gbl(1)
```

01.w_Defects/func_pointer.c:498

```
495 int flag;  
496 int (*func_gbl)(int);  
497 func_gbl = (int (*)(int))func_pointer_013_func_002;
```

```
498 flag = func_gbl(1);
```

```
499 }  
500  
501
```

```
func_gbl(1)
```

01.w_Defects/func_pointer.c:498

```
495 int flag;  
496 int (*func_gbl)(int);  
497 func_gbl = (int (*)(int))func_pointer_013_func_002;
```

```
498 flag = func_gbl(1);
```

```
499 }  
500  
501
```

01.w_Defects/func_pointer.c:538

Level Low

Status Not processed

```
535     if (flag == 1)
536     {
537         float f;
538         f = func_gbl();/*Tool should detect this line as error*/ /*ERROR:Bad function
pointer casting*/
539     }
540     return ret;
541 }
```

Trace

```
func_gbl()
```

01.w_Defects/func_pointer.c:538

```
535     if (flag == 1)
536     {
537         float f;
538         f = func_gbl();/*Tool should detect this line as error*/ /*ERROR:Bad function
pointer casting*/
539     }
540     return ret;
541 }
```

```
func_gbl()
```

01.w_Defects/func_pointer.c:538

```
535     if (flag == 1)
536     {
537         float f;
538         f = func_gbl();/*Tool should detect this line as error*/ /*ERROR:Bad function
pointer casting*/
539     }
540     return ret;
541 }
```

01.w_Defects/func_pointer.c:548

Level Low**Status** Not processed

```
545 int flag;
546 int (*fptr)(int);
547 fptr = func_pointer_014_func_002;
```

```
548 flag = fptr(1);
```

```
549 }
```

```
550
```

```
551 /*
```

Trace

fptr(1)

01.w_Defects/func_pointer.c:548

```
545 int flag;
546 int (*fptr)(int);
547 fptr = func_pointer_014_func_002;
```

```
548 flag = fptr(1);
```

```
549 }
```

```
550
```

```
551 /*
```

fptr(1)

01.w_Defects/func_pointer.c:548

```
545 int flag;
546 int (*fptr)(int);
547 fptr = func_pointer_014_func_002;
```

```
548 flag = fptr(1);
```

```
549 }
```

```
550
551 /*
```

01.w_Defects/func_pointer.c:592

Level Low

Status Not processed

```
589 else
590 {
591     fptr_gbl = (func_pointer_015_s_001 *) (func_pointer_015_s_001*)
func_pointer_015_func_003;

592     st = fptr_gbl( st1);/*Tool should detect this line as error*/ /*ERROR:Bad
function pointer casting*/

593 }
594 }
595
```

Trace

```
fptr_gbl( st1)
```

01.w_Defects/func_pointer.c:592

```
589 else
590 {
591     fptr_gbl = (func_pointer_015_s_001 *) (func_pointer_015_s_001*)
func_pointer_015_func_003;

592     st = fptr_gbl( st1);/*Tool should detect this line as error*/ /*ERROR:
Bad function pointer casting*/

593 }
594 }
595
```

```
fptr_gbl( st1)
```

01.w_Defects/func_pointer.c:592

```
589 else
590 {
591     fptr_gbl = (func_pointer_015_s_001 *) (func_pointer_015_s_001*)
func_pointer_015_func_003;

592     st = fptr_gbl( st1); /*Tool should detect this line as error*/ /*ERROR:
Bad function pointer casting*/

593 }
594 }
595
```

01.w_Defects/insign_code.c:24

Level Low

Status Not processed

```
21
22 i = rand();
23 j = i - 1;

24 i = j + 1; /*Tool should detect this line as error*/ /*ERROR:Useless Assignment */

25 }
26
27 /*
```

Trace

```
j + 1
```

01.w_Defects/insign_code.c:24

```
21
22 i = rand();
23 j = i - 1;
```

```
24    i = j + 1; /*Tool should detect this line as error*/ /*ERROR:Useless  
Assignment */
```

```
25 }  
26  
27 /*
```

j + 1

01.w_Defects/insign_code.c:24

```
21  
22    i = rand();  
23    j = i - 1;
```

```
24    i = j + 1; /*Tool should detect this line as error*/ /*ERROR:Useless  
Assignment */
```

```
25 }  
26  
27 /*
```

01.w_Defects/invalid_memory_access.c:45

Level Low

Status Not processed

```
42    }  
43    }  
44    if(flag == 10)
```

```
45        a = *(ptr+1); /*Tool should detect this line as error*/ /*ERROR:Invalid  
memory access to already freed area*/
```

```
46  
47 }  
48
```

Trace

```
*(ptr+1)
```

01.w_Defects/invalid_memory_access.c:45

```
42  }
43  }
44  if(flag == 10)

45          a = *(ptr+1); /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

46
47 }
48
```

```
*(ptr+1)
```

01.w_Defects/invalid_memory_access.c:45

```
42  }
43  }
44  if(flag == 10)

45          a = *(ptr+1); /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

46
47 }
48
```

01.w_Defects/invalid_memory_access.c:147

Level Low

Status Not processed

```
144
145 if (count ==0)
146 {
```

```
147          count = *ptr; /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/
```

```
148 return 1;  
149 }  
150 else
```

Trace

*ptr

01.w_Defects/invalid_memory_access.c:147

```
144  
145 if (count ==0)  
146 {
```

147 count = *ptr; /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

```
148 return 1;  
149 }  
150 else
```

*ptr

01.w_Defects/invalid_memory_access.c:147

```
144  
145 if (count ==0)  
146 {
```

147 count = *ptr; /*Tool should detect this line as error*/ /*ERROR:Invalid
memory access to already freed area*/

```
148 return 1;  
149 }  
150 else
```

01.w_Defects/invalid_memory_access.c:410

Level Low

Status Not processed

```
407 invalid_memory_access_013_s_001_s_gbl->a = 10;
408 invalid_memory_access_013_s_001_s_gbl->b = 10;
409 invalid_memory_access_013_s_001_s_gbl->uninit = 10;

410 i=invalid_memory_access_013_s_001_s_gbl->a;

411 free(invalid_memory_access_013_s_001_s_gbl);
412 }
413 break;
```

Trace

```
invalid_memory_access_013_s_001_s_gbl-
>a
```

01.w_Defects/invalid_memory_access.c:410

```
407 invalid_memory_access_013_s_001_s_gbl->a = 10;
408 invalid_memory_access_013_s_001_s_gbl->b = 10;
409 invalid_memory_access_013_s_001_s_gbl->uninit = 10;

410 i=invalid_memory_access_013_s_001_s_gbl->a;

411 free(invalid_memory_access_013_s_001_s_gbl);
412 }
413 break;
```

```
invalid_memory_access_013_s_001_s_gbl-
>a
```

01.w_Defects/invalid_memory_access.c:410

```
407 invalid_memory_access_013_s_001_s_gbl->a = 10;
408 invalid_memory_access_013_s_001_s_gbl->b = 10;
409 invalid_memory_access_013_s_001_s_gbl->uninit = 10;

410 i=invalid_memory_access_013_s_001_s_gbl->a;

411 free(invalid_memory_access_013_s_001_s_gbl);
```

```
412 }  
413 break;
```

01.w_Defects/invalid_memory_access.c:422

Level Low

Status Not processed

```
419 invalid_memory_access_013_s_001_s_gbl->a = 20;  
420 invalid_memory_access_013_s_001_s_gbl->b = 20;  
421 invalid_memory_access_013_s_001_s_gbl->uninit = 20;  
  
422 i=invalid_memory_access_013_s_001_s_gbl->a;  
  
423 free(invalid_memory_access_013_s_001_s_gbl);  
424 }  
425 break;
```

Trace

```
invalid_memory_access_013_s_001_s_gbl->a
```

01.w_Defects/invalid_memory_access.c:422

```
419 invalid_memory_access_013_s_001_s_gbl->a = 20;  
420 invalid_memory_access_013_s_001_s_gbl->b = 20;  
421 invalid_memory_access_013_s_001_s_gbl->uninit = 20;  
  
422 i=invalid_memory_access_013_s_001_s_gbl->a;  
  
423 free(invalid_memory_access_013_s_001_s_gbl);  
424 }  
425 break;
```

```
invalid_memory_access_013_s_001_s_gbl->a
```

01.w_Defects/invalid_memory_access.c:422

```
419 invalid_memory_access_013_s_001_s_gbl->a = 20;
420 invalid_memory_access_013_s_001_s_gbl->b = 20;
421 invalid_memory_access_013_s_001_s_gbl->uninit = 20;

422 i=invalid_memory_access_013_s_001_s_gbl->a;

423 free(invalid_memory_access_013_s_001_s_gbl);
424 }
425 break;
```

01.w_Defects/invalid_memory_access.c:532

Level Low

Status Not processed

```
529 {
530   {
531     char * str;

532     str = invalid_memory_access_015_func_001(buf[j]);

533   }
534 }
535 }
```

Trace

```
invalid_memory_access_015_func_001(buf[j])
```

01.w_Defects/invalid_memory_access.c:532

```
529 {
530   {
531     char * str;
```

```
532     str = invalid_memory_access_015_func_001(buf[j]);  
  
533 }  
534 }  
535 }
```

invalid_memory_access_015_func_001(buf
[j])

01.w_Defects/invalid_memory_access.c:532

```
529 {  
530 {  
531     char * str;  
  
532     str = invalid_memory_access_015_func_001(buf[j]);  
  
533 }  
534 }  
535 }
```

01.w_Defects/littlemem_st.c:36

Level Low

Status Not processed

```
33 }  
34  
35 p = (littlemem_st_001_s_001 *)buf;  
  
36 ret = p->c; /*Tool should detect this line as error*/ /*ERROR:Little Memory or  
Overflow*/  
  
37 printf("%d \n",p->c);  
38 }  
39
```

Trace

p->c

01.w_Defects/littlemem_st.c:36

```
33  }
34
35  p = (littlemem_st_001_s_001 *)buf;

36  ret = p->c; /*Tool should detect this line as error*/ /*ERROR:Little Memory or
Overflow*/

37  printf("%d \n",p->c);
38 }
39
```

p->c

01.w_Defects/littlemem_st.c:36

```
33  }
34
35  p = (littlemem_st_001_s_001 *)buf;

36  ret = p->c; /*Tool should detect this line as error*/ /*ERROR:Little Memory or
Overflow*/

37  printf("%d \n",p->c);
38 }
39
```

01.w_Defects/lock_never_unlock.c:228

Level Low**Status** Not processed

```
225
226  pthread_mutex_lock( &lock_never_unlock_004_glb_mutex_2 );
227  ip = (long)input;

228  ip = ip *20;
```

```
229
230 #if defined PRINT_DEBUG
231   printf("Task4_2! Lock Never Unlock, thread #%ld!\n",ip);
```

Trace

```
ip *20
```

01.w_Defects/lock_never_unlock.c:228

```
225
226   pthread_mutex_lock( &lock_never_unlock_004_glb_mutex_2 );
227   ip = (long)input;
```

```
228   ip = ip *20;
```

```
229
```

```
230 #if defined PRINT_DEBUG
231   printf("Task4_2! Lock Never Unlock, thread #%ld!\n",ip);
```

```
ip *20
```

01.w_Defects/lock_never_unlock.c:228

```
225
226   pthread_mutex_lock( &lock_never_unlock_004_glb_mutex_2 );
227   ip = (long)input;
```

```
228   ip = ip *20;
```

```
229
```

```
230 #if defined PRINT_DEBUG
231   printf("Task4_2! Lock Never Unlock, thread #%ld!\n",ip);
```

01.w_Defects/lock_never_unlock.c:325

Level Low

Status Not processed

```
322
323 pthread_mutex_lock( &lock_never_unlock_006_glb_mutex_2 );
324 ip = (long)input;

325 ip = ip *20;

326
327 #if defined PRINT_DEBUG
328 printf("Task6_2! Lock Never Unlock, thread #%ld!\n",ip);
```

Trace

```
ip *20
```

01.w_Defects/lock_never_unlock.c:325

```
322
323 pthread_mutex_lock( &lock_never_unlock_006_glb_mutex_2 );
324 ip = (long)input;

325 ip = ip *20;

326
327 #if defined PRINT_DEBUG
328 printf("Task6_2! Lock Never Unlock, thread #%ld!\n",ip);
```

```
ip *20
```

01.w_Defects/lock_never_unlock.c:325

```
322
323 pthread_mutex_lock( &lock_never_unlock_006_glb_mutex_2 );
324 ip = (long)input;

325 ip = ip *20;

326
327 #if defined PRINT_DEBUG
328 printf("Task6_2! Lock Never Unlock, thread #%ld!\n",ip);
```

01.w_Defects/memory_allocation_failure.c:282

Level Low**Status** Not processed

```
279     int i=0;
280     do
281     {
282         buf = (char*) malloc(MAX_BUFFER * sizeof(char));/*Tool should detect this
line as error*/ /*ERROR:Memory allocation failure */
283         i++;
284     }while (i<MAX_VAL);
285 }
```

Trace

(char*) malloc(MAX_BUFFER * sizeof(char))

01.w_Defects/memory_allocation_failure.c:282

```
279     int i=0;
280     do
281     {
282         buf = (char*) malloc(MAX_BUFFER * sizeof(char));/*Tool should
detect this line as error*/ /*ERROR:Memory allocation failure */
283         i++;
284     }while (i<MAX_VAL);
285 }
```

(char*) malloc(MAX_BUFFER * sizeof(char))

01.w_Defects/memory_allocation_failure.c:282

```
279     int i=0;
280     do
281     {
282         buf = (char*) malloc(MAX_BUFFER * sizeof(char));/*Tool should
```

detect this line as error*/ /*ERROR:Memory allocation failure */

```
283         i++;
284     }while (i<MAX_VAL);
285 }
```

01.w_Defects/memory_allocation_failure.c:448

Level Low

Status Not processed

```
445 if(flag == 10){
446     if(memory_allocation_failure_012_buf2_gbl!=NULL)
447     {
448         a = ptr[1][1];
449         free(memory_allocation_failure_012_buf2_gbl);
450     }
451 }
```

Trace

ptr[1][1]

01.w_Defects/memory_allocation_failure.c:448

```
445 if(flag == 10){
446     if(memory_allocation_failure_012_buf2_gbl!=NULL)
447     {
448         a = ptr[1][1];
449         free(memory_allocation_failure_012_buf2_gbl);
450     }
451 }
```

ptr[1][1]

01.w_Defects/memory_allocation_failure.c:448

```
445 if(flag == 10){  
446     if(memory_allocation_failure_012_buf2_gbl!=NULL)  
447     {  
  
448         a = ptr[1][1];  
  
449     free(memory_allocation_failure_012_buf2_gbl);  
450 }  
451 }
```

01.w_Defects/memory_leak.c:212

Level Low

Status Not processed

```
209 void memory_leak_008 ()  
210 {  
211     int *ptr=(int*) malloc(5 * sizeof(int));  
  
212     int *p = (int*)malloc(5 * sizeof(int));/*Tool should detect this line as error*/ /*ERROR:  
Memory Leakage */  
  
213     if(ptr !=NULL)  
214     {  
215         p = ptr;
```

Trace

(int*)malloc(5 * sizeof(int))

01.w_Defects/memory_leak.c:212

```
209 void memory_leak_008 ()  
210 {  
211     int *ptr=(int*) malloc(5 * sizeof(int));
```

```
212 int *p = (int*)malloc(5 * sizeof(int));/*Tool should detect this line as error*/
/*ERROR:Memory Leakage */
```

```
213 if(ptr !=NULL)
214 {
215 p = ptr;
```

(int*)malloc(5 * sizeof(int))

01.w_Defects/memory_leak.c:212

```
209 void memory_leak_008 ()
210 {
211 int *ptr=(int*) malloc(5 * sizeof(int));
```

```
212 int *p = (int*)malloc(5 * sizeof(int));/*Tool should detect this line as error*/
/*ERROR:Memory Leakage */
```

```
213 if(ptr !=NULL)
214 {
215 p = ptr;
```

01.w_Defects/memory_leak.c:228

Level Low

Status Not processed

```
225 void memory_leak_009 ()
226 {
227 float *ptr=(float*) malloc(5 * sizeof(float));
```

```
228 int *p = (int*) malloc(5 * sizeof(int)); /*Tool should detect this line as error*/
/*ERROR:Memory Leakage */
```

```
229 if(ptr !=NULL)
230 {
231     p = (int *)ptr;
```

Trace

```
(int*) malloc(5 * sizeof(int))
```

01.w_Defects/memory_leak.c:228

```
225 void memory_leak_009 ()  
226 {  
227   float *ptr=(float*) malloc(5 * sizeof(float));  
  
228   int *p = (int*) malloc(5 * sizeof(int)); /*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
  
229   if(ptr !=NULL)  
230   {  
231     p = (int *)ptr;
```

```
(int*) malloc(5 * sizeof(int))
```

01.w_Defects/memory_leak.c:228

```
225 void memory_leak_009 ()  
226 {  
227   float *ptr=(float*) malloc(5 * sizeof(float));  
  
228   int *p = (int*) malloc(5 * sizeof(int)); /*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
  
229   if(ptr !=NULL)  
230   {  
231     p = (int *)ptr;
```

01.w_Defects/memory_leak.c:245

Level Low

Status Not processed

```
242 void memory_leak_0010 ()  
243 {  
244   int *ptr = (int*) calloc(5,sizeof(int));
```

```
245   int *p1 = (int*) calloc(5,sizeof(int));/*Tool should detect this line as error*/ /*ERROR:  
Memory Leakage */
```

```
246 int *p2 = NULL;  
247 p1 = ptr;  
248 p2 = p1;
```

Trace

(int*) calloc(5,sizeof(int))

01.w_Defects/memory_leak.c:245

```
242 void memory_leak_0010 ()  
243 {  
244 int *ptr = (int*) calloc(5,sizeof(int));  
  
245 int *p1 = (int*) calloc(5,sizeof(int));/*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
  
246 int *p2 = NULL;  
247 p1 = ptr;  
248 p2 = p1;
```

(int*) calloc(5,sizeof(int))

01.w_Defects/memory_leak.c:245

```
242 void memory_leak_0010 ()  
243 {  
244 int *ptr = (int*) calloc(5,sizeof(int));  
  
245 int *p1 = (int*) calloc(5,sizeof(int));/*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
  
246 int *p2 = NULL;  
247 p1 = ptr;  
248 p2 = p1;
```

01.w_Defects/memory_leak.c:276

Level Low

Status Not processed

```
273 {  
274     char * buf ;  
275  
  
276     buf = un.u2;  
  
277 }  
278 }  
279
```

Trace

un.u2

01.w_Defects/memory_leak.c:276

```
273 {  
274     char * buf ;  
275  
  
276     buf = un.u2;  
  
277 }  
278 }  
279
```

un.u2

01.w_Defects/memory_leak.c:276

```
273 {  
274     char * buf ;  
275  
  
276     buf = un.u2;  
  
277 }  
278 }  
279
```

01.w_Defects/memory_leak.c:308

Level Low**Status** Not processed

```
305 void memory_leak_0012 ()  
306 {  
307     memory_leak_0012_uni_001 *u = (memory_leak_0012_uni_001 *)malloc(5*sizeof(  
memory_leak_0012_uni_001 ));  
  
308     memory_leak_0012_uni_001 *p = (memory_leak_0012_uni_001 *)malloc(5*sizeof(  
(memory_leak_0012_uni_001 ));/*Tool should detect this line as error*/ /*ERROR:  
Memory Leakage */  
  
309     p = u;  
310  
311     p->s1.a = 1;
```

Trace

```
(memory_leak_0012_uni_001 *)malloc  
(5*sizeof( memory_leak_0012_uni_001 ))
```

01.w_Defects/memory_leak.c:308

```
305 void memory_leak_0012 ()  
306 {  
307     memory_leak_0012_uni_001 *u = (memory_leak_0012_uni_001 *)malloc(5*sizeof(  
memory_leak_0012_uni_001 ));  
  
308     memory_leak_0012_uni_001 *p = (memory_leak_0012_uni_001 *)malloc(5*sizeof(  
(memory_leak_0012_uni_001 ));/*Tool should detect this line as error*/ /*ERROR:  
Memory Leakage */  
  
309     p = u;  
310  
311     p->s1.a = 1;
```

```
(memory_leak_0012_uni_001 * )malloc  
(5*sizeof( memory_leak_0012_uni_001 ))
```

01.w_Defects/memory_leak.c:308

```
305 void memory_leak_0012 ()  
306 {  
307     memory_leak_0012_uni_001 *u = (memory_leak_0012_uni_001 * )malloc  
(5*sizeof( memory_leak_0012_uni_001 ));  
  
308     memory_leak_0012_uni_001 *p = (memory_leak_0012_uni_001 * )malloc  
(5*sizeof( memory_leak_0012_uni_001 ));/*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
  
309     p = u;  
310  
311     p->s1.a = 1;
```

01.w_Defects/memory_leak.c:372

Level Low

Status Not processed

```
369 float **fp2 = &fptr;  
370 fptr = NULL;  
371 {  
  
372     float * fptr = *fp1;  
  
373     fptr = (float *)calloc(10, sizeof(float));/*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
374     if(fptr!=NULL)  
375     {
```

Trace

*fp1

01.w_Defects/memory_leak.c:372

```
369 float **fp2 = &fptr;  
370 fptr = NULL;  
371 {  
  
372     float * fptr = *fp1;  
  
373     fptr = (float *)calloc(10, sizeof(float));/*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
374     if(fptr!=NULL)  
375     {
```

*fp1

01.w_Defects/memory_leak.c:372

```
369 float **fp2 = &fptr;  
370 fptr = NULL;  
371 {  
  
372     float * fptr = *fp1;  
  
373     fptr = (float *)calloc(10, sizeof(float));/*Tool should detect this line as error*/  
/*ERROR:Memory Leakage */  
374     if(fptr!=NULL)  
375     {
```

01.w_Defects/memory_leak.c:382

Level Low

Status Not processed

```
379     }  
380     {  
381         float * fptr1 ;  
  
382         fptr1 = *fp2;
```

```
383 }
384 }
385
```

Trace

*fp2

01.w_Defects/memory_leak.c:382

```
379 }
380 {
381     float * fptr1 ;

382     fptr1 = *fp2;

383 }
```

*fp2

01.w_Defects/memory_leak.c:382

```
379 }
380 {
381     float * fptr1 ;

382     fptr1 = *fp2;

383 }
```

01.w_Defects/overrun_st.c:44

Level Low

Status Not processed

```
41 {  
42     int buf[5] = {1, 2, 3, 4, 5};  
43     int ret;  
  
44     ret = buf[5];/*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
45     sink = buf[idx];  
46 }  
47
```

Trace

buf[5]

01.w_Defects/overrun_st.c:44

```
41 {  
42     int buf[5] = {1, 2, 3, 4, 5};  
43     int ret;  
  
44     ret = buf[5];/*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
45     sink = buf[idx];  
46 }  
47
```

buf[5]

01.w_Defects/overrun_st.c:44

```
41 {  
42     int buf[5] = {1, 2, 3, 4, 5};  
43     int ret;  
  
44     ret = buf[5];/*Tool should detect this line as error*/ /*ERROR: buffer overrun */  
  
45     sink = buf[idx];  
46 }  
47
```

01.w_Defects/overrun_st.c:320

Level Low**Status** Not processed

```
317 int *p;  
318 int ret;  
319 p = buf;
```

```
320 ret = *(p + 5);/*Tool should detect this line as error*/ /*ERROR: buffer overrun */
```

```
321     sink = buf[idx];  
322 }  
323
```

Trace

*(p + 5)

01.w_Defects/overrun_st.c:320

```
317 int *p;  
318 int ret;  
319 p = buf;
```

```
320 ret = *(p + 5);/*Tool should detect this line as error*/ /*ERROR: buffer  
overrun */
```

```
321     sink = buf[idx];  
322 }  
323
```

*(p + 5)

01.w_Defects/overrun_st.c:320

```
317 int *p;  
318 int ret;  
319 p = buf;
```

```
320 ret = *(p + 5);/*Tool should detect this line as error*/ /*ERROR: buffer  
overrun */
```

```
321     sink = buf[idx];
322 }
323
```

01.w_Defects/ptr_subtraction.c:35

Level Low**Status** Not processed

```
32 int x= 10;
33 int *ptr = &x;
34 char *buf ;

35 buf= (char *)(ptr+1); /*Tool should detect this line as error*/ /*ERROR:Incorrect
pointer arithmetic*/

36 }
37
38 /*
```

Trace

(char *)(ptr+1)

01.w_Defects/ptr_subtraction.c:35

```
32 int x= 10;
33 int *ptr = &x;
34 char *buf ;

35 buf= (char *)(ptr+1); /*Tool should detect this line as error*/ /*ERROR:
Incorrect pointer arithmetic*/

36 }
37
38 /*
```

(char *)(ptr+1)

01.w_Defects/ptr_subtraction.c:35

```
32 int x= 10;
33 int *ptr = &x;
34 char *buf ;

35 buf= (char *)(ptr+1); /*Tool should detect this line as error*/ /*ERROR:
   Incorrect pointer arithmetic*/

36 }
37
38 /*
```

01.w_Defects/race_condition.c:372

Level Low

Status Not processed

```
369
370 /*pthread_mutex_lock( &race_condition_009_glb_mutex_1 );*/
371 ip = (long)input;

372 ip = ip *10;

373 race_condition_009_glb_data++; /*Tool should detect this line as error*/ /*ERROR:
   Race condition*/
374 #if defined PRINT_DEBUG
375 printf("Task4_1! race condition, thread #%ld!\n",ip);
```

Trace

ip *10

01.w_Defects/race_condition.c:372

```
369
370 /*pthread_mutex_lock( &race_condition_009_glb_mutex_1 );*/
371 ip = (long)input;
```

```
372 ip = ip *10;

373 race_condition_009_glb_data++; /*Tool should detect this line as error*/
/*ERROR:Race condition*/
374 #if defined PRINT_DEBUG
375 printf("Task4_1! race condition, thread #%ld!\n",ip);
```

ip *10

01.w_Defects/race_condition.c:372

```
369
370 /*pthread_mutex_lock( &race_condition_009_glb_mutex_1 );*/
371 ip = (long)input;

372 ip = ip *10;

373 race_condition_009_glb_data++; /*Tool should detect this line as error*/
/*ERROR:Race condition*/
374 #if defined PRINT_DEBUG
375 printf("Task4_1! race condition, thread #%ld!\n",ip);
```

01.w_Defects/race_condition.c:392

Level Low

Status Not processed

```
389
390 /*pthread_mutex_lock( &race_condition_009_glb_mutex_2 );*/
391 ip = (long)input;

392 ip = ip *20;

393 race_condition_009_glb_data--;
394 #if defined PRINT_DEBUG
395 printf("Task4_2! race condition, thread #%ld!\n",ip);
```

Trace

```
ip *20
```

01.w_Defects/race_condition.c:392

```
389
390 /*pthread_mutex_lock( &race_condition_009_glb_mutex_2 );*/
391 ip = (long)input;

392 ip = ip *20;

393 race_condition_009_glb_data--;
394 #if defined PRINT_DEBUG
395 printf("Task4_2! race condition, thread #%ld!\n",ip);
```

```
ip *20
```

01.w_Defects/race_condition.c:392

```
389
390 /*pthread_mutex_lock( &race_condition_009_glb_mutex_2 );*/
391 ip = (long)input;

392 ip = ip *20;

393 race_condition_009_glb_data--;
394 #if defined PRINT_DEBUG
395 printf("Task4_2! race condition, thread #%ld!\n",ip);
```

01.w_Defects/st_underrun.c:137

Level Low

Status Not processed

```
134 void st_underrun_004 ()
135 {
136   st_underrun_004_s_001 s,s2;

137   s2 = st_underrun_004_func_001(&s);

138 }
```

```
139
140 /*
```

Trace

```
st_underrun_004_func_001(&s)
```

01.w_Defects/st_underrun.c:137

```
134 void st_underrun_004 ()
135 {
136   st_underrun_004_s_001 s,s2;

137   s2 = st_underrun_004_func_001(&s);

138 }
139
140 /*
```

```
st_underrun_004_func_001(&s)
```

01.w_Defects/st_underrun.c:137

```
134 void st_underrun_004 ()
135 {
136   st_underrun_004_s_001 s,s2;

137   s2 = st_underrun_004_func_001(&s);

138 }
139
140 /*
```

01.w_Defects/st_underrun.c:195

Level Low

Status Not processed

```
192 char c;
```

```
193 for (;s.buf[len] != 'Z';len--) /*Tool should detect this line as error*/ /* Error: Stack  
Under RUN error */  
194 {  
  
195     c = s.buf[len];  
  
196 /*if (0)  
197 break;*/  
198 }
```

Trace

```
s.buf[len]
```

```
01.w_Defects/st_underrun.c:195
```

```
192 char c;  
193 for (;s.buf[len] != 'Z';len--) /*Tool should detect this line as error*/ /* Error:  
Stack Under RUN error */  
194 {  
  
195     c = s.buf[len];  
  
196 /*if (0)  
197 break;*/  
198 }
```

```
s.buf[len]
```

```
01.w_Defects/st_underrun.c:195
```

```
192 char c;  
193 for (;s.buf[len] != 'Z';len--) /*Tool should detect this line as error*/ /* Error:  
Stack Under RUN error */  
194 {  
  
195     c = s.buf[len];  
  
196 /*if (0)  
197 break;*/  
198 }
```

01.w_Defects/st_underrun.c:229

Level Low**Status** Not processed

```
226 char c;  
227 for (;s->buf[len] != 'Z';len--)  
228 {
```

```
229     c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under RUN error */
```

```
230 /*if (s->buf[len] == '\0')  
231     break;*/  
232 }
```

Trace

s->buf[len]

01.w_Defects/st_underrun.c:229

```
226 char c;  
227 for (;s->buf[len] != 'Z';len--)  
228 {
```

```
229     c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under RUN error */
```

```
230 /*if (s->buf[len] == '\0')  
231     break;*/  
232 }
```

s->buf[len]

01.w_Defects/st_underrun.c:229

```
226 char c;  
227 for (;s->buf[len] != 'Z';len--)  
228 {
```

```
229     c = s->buf[len]; /*Tool should detect this line as error*/ /* Stack Under
```

RUN error */

```
230 /*if (s->buf[len] == '\0')
231   break;*/
232 }
```

01.w_Defects/underrun_st.c:21

Level Low

Status Not processed

```
18 {
19   int buf[5] = {1, 2, 3, 4, 5};
20   int ret;

21   ret = buf[-1];/*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

22 }
23
24 /*
```

Trace

buf[-1]

01.w_Defects/underrun_st.c:21

```
18 {
19   int buf[5] = {1, 2, 3, 4, 5};
20   int ret;

21   ret = buf[-1];/*Tool should detect this line as error*/ /*ERROR:Data Underrun*/

22 }
23
24 /*
```

buf[-1]

01.w_Defects/underrun_st.c:21

```
18 {  
19     int buf[5] = {1, 2, 3, 4, 5};  
20     int ret;  
  
21     ret = buf[-1];/*Tool should detect this line as error*/ /*ERROR:Data  
Underrun*/  
  
22 }  
23  
24 /*
```

01.w_Defects/underrun_st.c:55

Level Low**Status** Not processed

```
52     int *p;  
53     int ret;  
54     p = buf;  
  
55     ret = *(p - 1);/*Tool should detect this line as error*/ /*ERROR:Data Underrun*/  
  
56 }  
57  
58 /*
```

Trace

*(p - 1)

01.w_Defects/underrun_st.c:55

```
52     int *p;  
53     int ret;  
54     p = buf;
```

```
55     ret = *(p - 1);/*Tool should detect this line as error*/ /*ERROR:Data Underrun*/
```

```
56 }
57
58 /*
```

```
*(p - 1)
```

01.w_Defects/underrun_st.c:55

```
52     int *p;
53     int ret;
54     p = buf;
```

```
55     ret = *(p - 1);/*Tool should detect this line as error*/ /*ERROR:Data Underrun*/
```

```
56 }
57
58 /*
```

01.w_Defects/uninit_memory_access.c:98

Level Low

Status Not processed

```
95
96             }
97         }
```

```
98     k = arr1[1][2][3];/*Tool should detect this line as error*/ /*ERROR:Uninitialized Memory Access*/
```

```
99 }
100
101 /*
```

Trace

arr1[1][2][3]

01.w_Defects/uninit_memory_access.c:98

```
95 }  
96 }  
97 }
```

98 k = arr1[1][2][3];/*Tool should detect this line as error*/ /*ERROR:
Uninitialized Memory Access*/

```
99 }  
100  
101 /*
```

arr1[1][2][3]

01.w_Defects/uninit_memory_access.c:98

```
95 }  
96 }  
97 }
```

98 k = arr1[1][2][3];/*Tool should detect this line as error*/ /*ERROR:
Uninitialized Memory Access*/

```
99 }  
100  
101 /*
```

01.w_Defects/uninit_memory_access.c:169

Level Low

Status Not processed

```
166 {  
167     int temp;  
168     if(num != 0) {
```

169 temp = num;

```
170 }
171 }
172 void uninit_memory_access_007 ()
```

Trace

```
temp = num
```

01.w_Defects/uninit_memory_access.c:169

```
166 {
167     int temp;
168     if(num != 0) {

169         temp = num;

170     }
171 }
172 void uninit_memory_access_007 ()
```

```
temp = num
```

01.w_Defects/uninit_memory_access.c:169

```
166 {
167     int temp;
168     if(num != 0) {

169         temp = num;

170     }
171 }
172 void uninit_memory_access_007 ()
```

01.w_Defects/uninit_memory_access.c:419

Level Low

Status Not processed

```
416 p = uninit_memory_access_014_func_001 ();
417 if(p != NULL)
418 {

419 ret = p->b; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Memory
Access*/

420 free(p);
421 p= NULL;
422 }
```

Trace

p->b

01.w_Defects/uninit_memory_access.c:419

```
416 p = uninit_memory_access_014_func_001 ();
417 if(p != NULL)
418 {

419 ret = p->b; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Memory Access*/

420 free(p);
421 p= NULL;
422 }
```

p->b

01.w_Defects/uninit_memory_access.c:419

```
416 p = uninit_memory_access_014_func_001 ();
417 if(p != NULL)
418 {

419 ret = p->b; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Memory Access*/

420 free(p);
421 p= NULL;
422 }
```

01.w_Defects/uninit_pointer.c:30

Level Low**Status** Not processed

```
27 int a = 5;
28 int *p ;
29 int ret;

30 ret = *p; /*Tool should detect this line as error*/ /*ERROR:Uninitialized pointer*/

31 }
32
33 /*
```

Trace

*p

01.w_Defects/uninit_pointer.c:30

```
27 int a = 5;
28 int *p ;
29 int ret;

30 ret = *p; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/

31 }
32
33 /*
```

*p

01.w_Defects/uninit_pointer.c:30

```
27 int a = 5;
28 int *p ;
29 int ret;

30 ret = *p; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/
```

```
31 }  
32  
33 /*
```

01.w_Defects/uninit_pointer.c:55

Level Low**Status** Not processed

```
52 int a = 0;  
53 int ret;  
54 pp = &p;  
  
55 ret = **pp; /*Tool should detect this line as error*/ /*ERROR:Uninitialized pointer*/  
  
56 }  
57  
58 /*
```

Trace

**pp

01.w_Defects/uninit_pointer.c:55

```
52 int a = 0;  
53 int ret;  
54 pp = &p;  
  
55 ret = **pp; /*Tool should detect this line as error*/ /*ERROR:Uninitialized  
pointer*/  
  
56 }  
57  
58 /*
```

```
**pp
```

01.w_Defects/uninit_pointer.c:55

```
52 int a = 0;
53 int ret;
54 pp = &p;

55 ret = **pp; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/

56 }
57
58 /*
```

01.w_Defects/uninit_pointer.c:65

Level Low

Status Not processed

```
62 void uninit_pointer_004_func_001 (int *p)
63 {
64     int ret;

65     ret = 0;

66 }
67 void uninit_pointer_004 ()
68 {
```

Trace

```
ret = 0
```

01.w_Defects/uninit_pointer.c:65

```
62 void uninit_pointer_004_func_001 (int *p)
63 {
64     int ret;
```

```
65     ret = 0;  
  
66 }  
67 void uninit_pointer_004 ()  
68 {
```

ret = 0

01.w_Defects/uninit_pointer.c:65

```
62 void uninit_pointer_004_func_001 (int *p)  
63 {  
64     int ret;  
  
65     ret = 0;  
  
66 }  
67 void uninit_pointer_004 ()  
68 {
```

01.w_Defects/uninit_pointer.c:81

Level Low

Status Not processed

```
78 void uninit_pointer_005_func_001 (int *pbuf[])  
79 {  
80     int buf1[6] = {1, 2, 3, 4, 5, 6};  
  
81     int buf2[6] = {1, 2, 3, 4, 5, 6};  
  
82     int buf3[6] = {1, 2, 3, 4, 5, 6};  
83     int buf4[6] = {1, 2, 3, 4, 5, 6};  
84     int buf5[6] = {1, 2, 3, 4, 5, 6};
```

Trace

{1, 2, 3, 4, 5, 6}

01.w_Defects/uninit_pointer.c:81

```
78 void uninit_pointer_005_func_001 (int *pbuff[])
79 {
80     int buf1[6] = {1, 2, 3, 4, 5, 6};

81     int buf2[6] = {1, 2, 3, 4, 5, 6};

82     int buf3[6] = {1, 2, 3, 4, 5, 6};
83     int buf4[6] = {1, 2, 3, 4, 5, 6};
84     int buf5[6] = {1, 2, 3, 4, 5, 6};
```

{1, 2, 3, 4, 5, 6}

01.w_Defects/uninit_pointer.c:81

```
78 void uninit_pointer_005_func_001 (int *pbuff[])
79 {
80     int buf1[6] = {1, 2, 3, 4, 5, 6};

81     int buf2[6] = {1, 2, 3, 4, 5, 6};

82     int buf3[6] = {1, 2, 3, 4, 5, 6};
83     int buf4[6] = {1, 2, 3, 4, 5, 6};
84     int buf5[6] = {1, 2, 3, 4, 5, 6};
```

01.w_Defects/uninit_pointer.c:90

Level Low**Status** Not processed

```
87     pbuf[3] = buf4;
88     pbuf[4] = buf5;
89     int ret;

90     ret = pbuf[1][1];

91 }
```

```
92 void uninit_pointer_005 ()  
93 {
```

Trace

pbuff[1][1]

01.w_Defects/uninit_pointer.c:90

```
87    pbuf[3] = buf4;  
88    pbuf[4] = buf5;  
89    int ret;  
  
90    ret = pbuf[1][1];  
  
91 }  
92 void uninit_pointer_005 ()  
93 {
```

pbuff[1][1]

01.w_Defects/uninit_pointer.c:90

```
87    pbuf[3] = buf4;  
88    pbuf[4] = buf5;  
89    int ret;  
  
90    ret = pbuf[1][1];  
  
91 }  
92 void uninit_pointer_005 ()  
93 {
```

01.w_Defects/uninit_pointer.c:200

Level Low

Status Not processed

```
197 /* cast void pointer to a pointer of the appropriate type */
```

```
198     char * * cptr = (char * *)vptr;
199     char * buf;

200     buf = (*cptr);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/

201 }
202 void uninit_pointer_010 ()
203 {
```

Trace

(*cptr)

01.w_Defects/uninit_pointer.c:200

```
197     /* cast void pointer to a pointer of the appropriate type */
198     char * * cptr = (char * *)vptr;
199     char * buf;

200     buf = (*cptr);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/

201 }
202 void uninit_pointer_010 ()
203 {
```

(*cptr)

01.w_Defects/uninit_pointer.c:200

```
197     /* cast void pointer to a pointer of the appropriate type */
198     char * * cptr = (char * *)vptr;
199     char * buf;

200     buf = (*cptr);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/

201 }
202 void uninit_pointer_010 ()
203 {
```

01.w_Defects/uninit_pointer.c:335

Level Low**Status** Not processed

```
332 uninit_pointer_014_func_001 (1);
333 if(s!=NULL)
334 {
335     r = *s; /*Tool should detect this line as error*/ /*ERROR:Uninitialized pointer*/
336     free(s);
337 }
338 }
```

Trace

*s

01.w_Defects/uninit_pointer.c:335

```
332 uninit_pointer_014_func_001 (1);
333 if(s!=NULL)
334 {
335     r = *s; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/
336     free(s);
337 }
338 }
```

*s

01.w_Defects/uninit_pointer.c:335

```
332 uninit_pointer_014_func_001 (1);
333 if(s!=NULL)
334 {
335     r = *s; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
pointer*/
```

```
336     free(s);
337 }
338 }
```

01.w_Defects/uninit_var.c:23

Level Low

Status Not processed

```
20 {
21     int a ;
22     int ret;

23     ret = a; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/

24 }
25
26 /*
```

Trace

```
ret = a; /*Tool should detect this line as
error*/ /*ERROR:Uninitialized Variable*/
```

01.w_Defects/uninit_var.c:23

```
20 {
21     int a ;
22     int ret;

23     ret = a; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
Variable*/

24 }
25
26 /*
```

```
ret = a; /*Tool should detect this line as  
error*/ /*ERROR:Uninitialized Variable*/
```

01.w_Defects/uninit_var.c:23

```
20 {  
21   int a ;  
22   int ret;  
  
23   ret = a; /*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/  
  
24 }  
25  
26 /*
```

01.w_Defects/uninit_var.c:34

Level Low

Status Not processed

```
31 {  
32   int buf[5];  
33   int ret;  
  
34   ret = buf[3]; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/  
  
35 }  
36  
37 /*
```

Trace

buf[3]

01.w_Defects/uninit_var.c:34

```
31 {  
32   int buf[5];  
33   int ret;
```

```
34     ret = buf[3];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/
```

```
35 }  
36  
37 /*
```

buf[3]

01.w_Defects/uninit_var.c:34

```
31 {  
32     int buf[5];  
33     int ret;
```

```
34     ret = buf[3];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/
```

```
35 }  
36  
37 /*
```

01.w_Defects/uninit_var.c:45

Level Low

Status Not processed

```
42 {  
43     int buf[5][6];  
44     int ret;
```

```
45     ret = buf[1][1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/
```

```
46 }  
47  
48 /*
```

Trace

buf[1][1]

01.w_Defects/uninit_var.c:45

```
42 {  
43     int buf[5][6];  
44     int ret;  
  
45     ret = buf[1][1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/  
  
46 }  
47  
48 /*
```

buf[1][1]

01.w_Defects/uninit_var.c:45

```
42 {  
43     int buf[5][6];  
44     int ret;  
  
45     ret = buf[1][1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/  
  
46 }  
47  
48 /*
```

01.w_Defects/uninit_var.c:63

Level Low

Status Not processed

```
60         dvar1 = 25.8;  
61     else  
62         ;
```

63 ret = dvar; /*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/

```
64 }  
65  
66 /*
```

Trace

```
ret = dvar; /*Tool should detect this line as  
error*/ /*ERROR:Uninitialized Variable*/
```

01.w_Defects/uninit_var.c:63

```
60      dvar1 = 25.8;  
61  else  
62      ;
```

```
63  ret = dvar; /*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/
```

```
64 }  
65  
66 /*
```

```
ret = dvar; /*Tool should detect this line as  
error*/ /*ERROR:Uninitialized Variable*/
```

01.w_Defects/uninit_var.c:63

```
60      dvar1 = 25.8;  
61  else  
62      ;
```

```
63  ret = dvar; /*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Variable*/
```

```
64 }  
65  
66 /*
```

01.w_Defects/uninit_var.c:81

Level Low

Status Not processed

```
78 void uninit_var_005 ()  
79 {  
80     int a;  
  
81     a = uninit_var_005_func_001();  
  
82 }  
83  
84 /*
```

Trace

uninit_var_005_func_001()

01.w_Defects/uninit_var.c:81

```
78 void uninit_var_005 ()  
79 {  
80     int a;  
  
81     a = uninit_var_005_func_001();  
  
82 }  
83  
84 /*
```

uninit_var_005_func_001()

01.w_Defects/uninit_var.c:81

```
78 void uninit_var_005 ()  
79 {  
80     int a;  
  
81     a = uninit_var_005_func_001();  
  
82 }  
83  
84 /*
```

01.w_Defects/uninit_var.c:111

Level Low**Status** Not processed

```
108 int ret;
109 s.a = 1;
110 s.b = 1;
```

```
111 ret = s.uninit; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
Variable*/
```

```
112 }
113
114 /*
```

Trace

s.uninit

01.w_Defects/uninit_var.c:111

```
108 int ret;
109 s.a = 1;
110 s.b = 1;
```

```
111 ret = s.uninit; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
Variable*/
```

```
112 }
113
114 /*
```

s.uninit

01.w_Defects/uninit_var.c:111

```
108 int ret;
109 s.a = 1;
110 s.b = 1;

111 ret = s.uninit; /*Tool should detect this line as error*/ /*ERROR:Uninitialized
Variable*/

112 }
113
114 /*
```

01.w_Defects/uninit_var.c:186

Level Low**Status** Not processed

```
183 {
184 int arr[5];
185 int p ;

186 p= uninit_var_011_func_001(arr,(sizeof(arr)/sizeof(int)));

187 }
188
189 /*
```

Trace

uninit_var_011_func_001(arr,(sizeof(arr)
/sizeof(int)))

01.w_Defects/uninit_var.c:186

```
183 {
184 int arr[5];
185 int p ;
```

```
186 p= uninit_var_011_func_001(arr,(sizeof(arr)/sizeof(int)));
```

```
187 }
```

```
188
```

```
189 /*
```

```
uninit_var_011_func_001(arr,(sizeof(arr)
/sizeof(int)))
```

01.w_Defects/uninit_var.c:186

```
183 {
```

```
184 int arr[5];
```

```
185 int p ;
```

```
186 p= uninit_var_011_func_001(arr,(sizeof(arr)/sizeof(int)));
```

```
187 }
```

```
188
```

```
189 /*
```

01.w_Defects/uninit_var.c:228

Level Low

Status Not processed

```
225 int ret;
```

```
226 ret = uninit_var_012_func_001 (s);
```

```
227 if(ret >=0)
```

```
228 r = s;
```

```
229 }
```

```
230
```

```
231 /*
```

Trace

```
r = s
```

01.w_Defects/uninit_var.c:228

```
225 int ret;
226 ret = uninit_var_012_func_001 (s);
227 if(ret >=0)

228         r = s;

229 }
230
231 /*
```

```
r = s
```

01.w_Defects/uninit_var.c:228

```
225 int ret;
226 ret = uninit_var_012_func_001 (s);
227 if(ret >=0)

228         r = s;

229 }
230
231 /*
```

01.w_Defects/uninit_var.c:249

Level Low

Status Not processed

```
246 void uninit_var_013 ()
247 {
248     int a;

249     a = uninit_var_013_func_001();

250 }
```

```
251
252 /*
```

Trace

uninit_var_013_func_001()

01.w_Defects/uninit_var.c:249

```
246 void uninit_var_013 ()
247 {
248     int a;

249     a = uninit_var_013_func_001();

250 }
251
252 /*
```

uninit_var_013_func_001()

01.w_Defects/uninit_var.c:249

```
246 void uninit_var_013 ()
247 {
248     int a;

249     a = uninit_var_013_func_001();

250 }
251
252 /*
```

01.w_Defects/uninit_var.c:276

Level Low

Status Not processed

```
273 uninit_var_014_s_001 s,r;
```

```
274 s.a = 1;  
275 s.b = 1;  
  
276 r = uninit_var_014_func_001(s);  
  
277 }  
278  
279 /*
```

Trace

```
uninit_var_014_func_001(s)
```

01.w_Defects/uninit_var.c:276

```
273 uninit_var_014_s_001 s,r;  
274 s.a = 1;  
275 s.b = 1;  
  
276 r = uninit_var_014_func_001(s);  
  
277 }  
278  
279 /*
```

```
uninit_var_014_func_001(s)
```

01.w_Defects/uninit_var.c:276

```
273 uninit_var_014_s_001 s,r;  
274 s.a = 1;  
275 s.b = 1;  
  
276 r = uninit_var_014_func_001(s);  
  
277 }  
278  
279 /*
```

01.w_Defects/uninit_var.c:296

Level Low**Status** Not processed

```
293 {  
294     int a[3],ret;  
295     uninit_var_015_func_001(a);  
  
296     ret = a[1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/  
  
297 };  
298  
299
```

Trace

a[1]

01.w_Defects/uninit_var.c:296

```
293 {  
294     int a[3],ret;  
295     uninit_var_015_func_001(a);  
  
296     ret = a[1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/  
  
297 };  
298  
299
```

a[1]

01.w_Defects/uninit_var.c:296

```
293 {  
294     int a[3],ret;  
295     uninit_var_015_func_001(a);  
  
296     ret = a[1];/*Tool should detect this line as error*/ /*ERROR:Uninitialized Variable*/
```

```
297 };
298
299
```

01.w_Defects/unused_var.c:24

Level Low

Status Not processed

```
21 int b = 2;
22 int unuse;
23

24 unuse = a + b; /*Tool should detect this line as error*/ /*ERROR:Unused variable*/

25 }
26
27 /*
```

Trace

a + b

01.w_Defects/unused_var.c:24

```
21 int b = 2;
22 int unuse;
23
```

```
24 unuse = a + b; /*Tool should detect this line as error*/ /*ERROR:Unused
variable*/
```

```
25 }
26
27 /*
```

a + b

01.w_Defects/unused_var.c:24

```
21 int b = 2;
22 int unuse;
23
24 unuse = a + b; /*Tool should detect this line as error*/ /*ERROR:Unused
variable*/
25 }
26
27 /*
```

01.w_Defects/wrong_arguments_func_pointer.c:53

Level Low**Status** Not processed

```
50 int (*fptr)(int *);
51 int a;
52 fptr = (int (*)(int *))wrong_arguments_func_pointer_001_func_001;
53 a =fptr(arr);/*Tool should detect this line as error__*//*ERROR:Wrong arguments
passed to a function pointer*/
54 }
55
56 /*
```

Trace

fptr(arr)

01.w_Defects/wrong_arguments_func_pointer.c:53

```
50 int (*fptr)(int *);
51 int a;
52 fptr = (int (*)(int *))wrong_arguments_func_pointer_001_func_001;
```

```
53 a =fptr(arr);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
54 }  
55  
56 /*
```

fptr(arr)

01.w_Defects/wrong_arguments_func_pointer.c:53

```
50 int (*fptr)(int *);  
51 int a;  
52 fptr = (int (*)(int *))wrong_arguments_func_pointer_001_func_001;
```

```
53 a =fptr(arr);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
54 }  
55  
56 /*
```

01.w_Defects/wrong_arguments_func_pointer.c:74

Level Low

Status Not processed

```
71 int (*fptr)(char *);  
72 int a;  
73 fptr = (int (*)(char *))wrong_arguments_func_pointer_002_func_001;
```

```
74 a =fptr(buf);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
75 }  
76  
77 /*
```

Trace

fptr(buf)

01.w_Defects/wrong_arguments_func_pointer.c:74

```
71 int (*fptr)(char *);
72 int a;
73 fptr = (int (*)(char *))wrong_arguments_func_pointer_002_func_001;

74 a =fptr(buf);/*Tool should detect this line as error__*//*ERROR:Wrong
arguments passed to a function pointer*/

75 }
76
77 /*
```

fptr(buf)

01.w_Defects/wrong_arguments_func_pointer.c:74

```
71 int (*fptr)(char *);
72 int a;
73 fptr = (int (*)(char *))wrong_arguments_func_pointer_002_func_001;

74 a =fptr(buf);/*Tool should detect this line as error__*//*ERROR:Wrong
arguments passed to a function pointer*/

75 }
76
77 /*
```

01.w_Defects/wrong_arguments_func_pointer.c:94

Level Low

Status Not processed

```
91 int a = 1;
92 int ret;
93 func = (int (*)(int))wrong_arguments_func_pointer_003_func_001;
```

```
94 ret = func(a);/*Tool should detect this line as error__*//*ERROR:Wrong arguments
passed to a function pointer*/
```

```
95 }  
96  
97 /*
```

Trace

```
func(a)
```

01.w_Defects/wrong_arguments_func_pointer.c:94

```
91 int a = 1;  
92 int ret;  
93 func = (int (*)(int))wrong_arguments_func_pointer_003_func_001;  
  
94 ret = func(a);/*Tool should detect this line as error**/*ERROR:Wrong  
arguments passed to a function pointer*/  
  
95 }  
96  
97 /*
```

```
func(a)
```

01.w_Defects/wrong_arguments_func_pointer.c:94

```
91 int a = 1;  
92 int ret;  
93 func = (int (*)(int))wrong_arguments_func_pointer_003_func_001;  
  
94 ret = func(a);/*Tool should detect this line as error**/*ERROR:Wrong  
arguments passed to a function pointer*/  
  
95 }  
96  
97 /*
```

01.w_Defects/wrong_arguments_func_pointer.c:114

Level Low

Status Not processed

```
111 char ret;
112 float a =20.5;
113 func = (char (*)(float ))wrong_arguments_func_pointer_004_func_001;
```

114 ret = func(a);/*Tool should detect this line as error**/*ERROR:Wrong arguments passed to a function pointer*/

```
115
116 }
117
```

Trace

```
func(a)
```

01.w_Defects/wrong_arguments_func_pointer.c:114

```
111 char ret;
112 float a =20.5;
113 func = (char (*)(float ))wrong_arguments_func_pointer_004_func_001;
```

114 ret = func(a);/*Tool should detect this line as error**/*ERROR:Wrong arguments passed to a function pointer*/

```
115
116 }
117
```

```
func(a)
```

01.w_Defects/wrong_arguments_func_pointer.c:114

```
111 char ret;
112 float a =20.5;
113 func = (char (*)(float ))wrong_arguments_func_pointer_004_func_001;
```

114 ret = func(a);/*Tool should detect this line as error**/*ERROR:Wrong arguments passed to a function pointer*/

```
115  
116 }  
117
```

01.w_Defects/wrong_arguments_func_pointer.c:141

Level Low

Status Not processed

```
138 float *buf = &i;  
139 float ret;  
140 func = (float (*)(float *))wrong_arguments_func_pointer_005_func_001;
```

141 ret = func(buf);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/

```
142  
143 }  
144
```

Trace

func(buf)

01.w_Defects/wrong_arguments_func_pointer.c:141

```
138 float *buf = &i;  
139 float ret;  
140 func = (float (*)(float *))wrong_arguments_func_pointer_005_func_001;
```

141 ret = func(buf);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/

```
142  
143 }  
144
```

```
func(buf)
```

01.w_Defects/wrong_arguments_func_pointer.c:141

```
138 float *buf = &i;  
139 float ret;  
140 func = (float (*)(float *))wrong_arguments_func_pointer_005_func_001;  
  
141 ret = func(buf);/*Tool should detect this line as error**/*ERROR:Wrong  
arguments passed to a function pointer*/  
  
142  
143 }  
144
```

01.w_Defects/wrong_arguments_func_pointer.c:161

Level Low

Status Not processed

```
158 int (*func)(int);  
159 int ret;  
160 func = (int (*)(int))wrong_arguments_func_pointer_006_func_001;  
  
161 ret = func(5);/*Tool should detect this line as error**/*ERROR:Wrong arguments  
passed to a function pointer*/  
  
162  
163 }  
164
```

Trace

```
func(5)
```

01.w_Defects/wrong_arguments_func_pointer.c:161

```
158 int (*func)(int);  
159 int ret;  
160 func = (int (*)(int))wrong_arguments_func_pointer_006_func_001;
```

```
161 ret = func(5);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
162  
163 }  
164
```

```
func(5)
```

01.w_Defects/wrong_arguments_func_pointer.c:161

```
158 int (*func)(int);  
159 int ret;  
160 func = (int (*)(int))wrong_arguments_func_pointer_006_func_001;
```

```
161 ret = func(5);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
162  
163 }  
164
```

01.w_Defects/wrong_arguments_func_pointer.c:182

Level Low

Status Not processed

```
179 unsigned int (*func)(double, double);  
180 unsigned int ret;  
181 func = (unsigned int (*)(double,double))  
wrong_arguments_func_pointer_007_func_001;
```

```
182 ret = func(1.005, 2.005);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
183 }  
184  
185 /*
```

Trace

```
func(1.005, 2.005)
```

01.w_Defects/wrong_arguments_func_pointer.c:182

```
179 unsigned int (*func)(double, double);
180 unsigned int ret;
181 func = (unsigned int (*)(double,double))
wrong_arguments_func_pointer_007_func_001;
```

182 ret = func(1.005, 2.005);/*Tool should detect this line as error__*/
Wrong arguments passed to a function pointer*/

```
183 }
184
185 /*
```

```
func(1.005, 2.005)
```

01.w_Defects/wrong_arguments_func_pointer.c:182

```
179 unsigned int (*func)(double, double);
180 unsigned int ret;
181 func = (unsigned int (*)(double,double))
wrong_arguments_func_pointer_007_func_001;
```

182 ret = func(1.005, 2.005);/*Tool should detect this line as error__*/
Wrong arguments passed to a function pointer*/

```
183 }
184
185 /*
```

01.w_Defects/wrong_arguments_func_pointer.c:201

Level Low

Status Not processed

```
198 char a = 'a',b='b';
199 float ret;
200 func_glb = (float (*)(char *,char *))wrong_arguments_func_pointer_008_func_001;
```

```
201 ret = func_glb(&a,&b);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
202  
203 }  
204
```

Trace

```
func_glb(&a,&b)
```

```
01.w_Defects/wrong_arguments_func_pointer.c:201
```

```
198 char a = 'a',b='b';  
199 float ret;  
200 func_glb = (float (*)(char *,char *))  
wrong_arguments_func_pointer_008_func_001;
```

```
201 ret = func_glb(&a,&b);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
202  
203 }  
204
```

```
func_glb(&a,&b)
```

```
01.w_Defects/wrong_arguments_func_pointer.c:201
```

```
198 char a = 'a',b='b';  
199 float ret;  
200 func_glb = (float (*)(char *,char *))  
wrong_arguments_func_pointer_008_func_001;
```

```
201 ret = func_glb(&a,&b);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
202  
203 }  
204
```

01.w_Defects/wrong_arguments_func_pointer.c:225

Level Low**Status** Not processed

```
222 char ret;
223 char (*func)(char ,char , int *);
224 func = (char (*)(char ,char,int*))wrong_arguments_func_pointer_009_func_001;

225 ret = func(*str1,*str2,str3);/*Tool should detect this line as error**/*ERROR:Wrong
arguments passed to a function pointer*/

226
227 }
228
```

Trace

func(*str1,*str2,str3)

01.w_Defects/wrong_arguments_func_pointer.c:225

```
222 char ret;
223 char (*func)(char ,char , int *);
224 func = (char (*)(char ,char,int*))wrong_arguments_func_pointer_009_func_001;
```

```
225 ret = func(*str1,*str2,str3);/*Tool should detect this line as error**/*ERROR:
Wrong arguments passed to a function pointer*/
```

```
226
227 }
228
```

func(*str1,*str2,str3)

01.w_Defects/wrong_arguments_func_pointer.c:225

```
222 char ret;
223 char (*func)(char ,char , int *);
224 func = (char (*)(char ,char,int*))wrong_arguments_func_pointer_009_func_001;
```

```
225 ret = func(*str1,*str2,str3);/*Tool should detect this line as error*//*ERROR:  
Wrong arguments passed to a function pointer*/
```

```
226  
227 }  
228
```

01.w_Defects/wrong_arguments_func_pointer.c:282

Level Low

Status Not processed

```
279 for (i = 0; i < MAX; i++)  
280 {  
281     st->arr[i] = i;  
  
282     temp = st->arr[i];  
  
283 }  
284 }  
285
```

Trace

st->arr[i]

01.w_Defects/wrong_arguments_func_pointer.c:282

```
279 for (i = 0; i < MAX; i++)  
280 {  
281     st->arr[i] = i;  
  
282     temp = st->arr[i];  
  
283 }  
284 }  
285
```

st->arr[i]

01.w_Defects/wrong_arguments_func_pointer.c:282

```
279     for (i = 0; i < MAX; i++)  
280     {  
281         st->arr[i] = i;  
  
282         temp = st->arr[i];  
  
283     }  
284 }  
285
```

01.w_Defects/wrong_arguments_func_pointer.c:423

Level Low**Status** Not processed

```
420     long (*fptr)(float *);  
421     long a;  
422     fptr = (long (*)(float * ))wrong_arguments_func_pointer_014_func_002;  
  
423     a =fptr(&f);/*Tool should detect this line as error**/*ERROR:Wrong  
arguments passed to a function pointer*/  
  
424 }  
425  
426 }
```

Trace

fptr(&f)

01.w_Defects/wrong_arguments_func_pointer.c:423

```
420     long (*fptr)(float *);  
421     long a;  
422     fptr = (long (*)(float * ))  
wrong_arguments_func_pointer_014_func_002;
```

```
423     a =fptr(&f);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
424 }
425
426 }
```

fptr(&f)

01.w_Defects/wrong_arguments_func_pointer.c:423

```
420     long (*fptr)(float *);
421     long a;
422     fptr = (long (*)(float * ))
wrong_arguments_func_pointer_014_func_002;
```

```
423     a =fptr(&f);/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
424 }
425
426 }
```

01.w_Defects/wrong_arguments_func_pointer.c:491

Level Low

Status Not processed

```
488 char ret;
489 char (*func)(char *,float *,int * );
490 func = (char (*)(char*,float*,int*))wrong_arguments_func_pointer_016_func_001;
/*Tool should detect this line as error__*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
491 ret = func(str1,str3,str2);
```

```
492 }
493
494 /*
```

Trace

```
func(str1,str3,str2)
```

01.w_Defects/wrong_arguments_func_pointer.c:491

```
488 char ret;
489 char (*func)(char *,float *,int * );
490 func = (char (*)(char*,float*,int*))  
wrong_arguments_func_pointer_016_func_001; /*Tool should detect this line as  
error*/ /*ERROR:Wrong arguments passed to a function pointer*/  
  
491 ret = func(str1,str3,str2);  
  
492 }
493
494 /*
```

```
func(str1,str3,str2)
```

01.w_Defects/wrong_arguments_func_pointer.c:491

```
488 char ret;
489 char (*func)(char *,float *,int * );
490 func = (char (*)(char*,float*,int*))  
wrong_arguments_func_pointer_016_func_001; /*Tool should detect this line as  
error*/ /*ERROR:Wrong arguments passed to a function pointer*/  
  
491 ret = func(str1,str3,str2);  
  
492 }
493
494 /*
```

01.w_Defects/wrong_arguments_func_pointer.c:502

Level Low

Status Not processed

```
499 int wrong_arguments_func_pointer_017_func_001 (int flag,float flag2)
500 {
501 float a=0.0;
```

```
502 a += flag2;
```

```
503 flag = 1;  
504 return flag;  
505 }
```

Trace

```
a += flag2
```

```
01.w_Defects/wrong_arguments_func_pointer.c:502
```

```
499 int wrong_arguments_func_pointer_017_func_001 (int flag,float flag2)  
500 {  
501 float a=0.0;
```

```
502 a += flag2;
```

```
503 flag = 1;  
504 return flag;  
505 }
```

```
a += flag2
```

```
01.w_Defects/wrong_arguments_func_pointer.c:502
```

```
499 int wrong_arguments_func_pointer_017_func_001 (int flag,float flag2)  
500 {  
501 float a=0.0;
```

```
502 a += flag2;
```

```
503 flag = 1;  
504 return flag;  
505 }
```

```
01.w_Defects/wrong_arguments_func_pointer.c:528
```

Level Low

Status Not processed

```
525 my_label2:  
526     if (flag == 1)  
527 {
```

```
528     flag = wrong_arguments_func_pointer_017_func_gbl(1.9,0);/*Tool should  
detect this line as error*//*ERROR:Wrong arguments passed to a function pointer*/
```

```
529     flag2++;  
530 }  
531 return ret;
```

Trace

```
wrong_arguments_func_pointer_017_func_  
gbl(1.9,0)
```

01.w_Defects/wrong_arguments_func_pointer.c:528

```
525 my_label2:  
526     if (flag == 1)  
527 {
```

```
528     flag = wrong_arguments_func_pointer_017_func_gbl(1.9,0);/*Tool  
should detect this line as error*//*ERROR:Wrong arguments passed to a function  
pointer*/
```

```
529     flag2++;  
530 }  
531 return ret;
```

```
wrong_arguments_func_pointer_017_func_  
gbl(1.9,0)
```

01.w_Defects/wrong_arguments_func_pointer.c:528

```
525 my_label2:  
526     if (flag == 1)  
527 {
```

```
528     flag = wrong_arguments_func_pointer_017_func_gbl(1.9,0);/*Tool
```

should detect this line as error/*ERROR:Wrong arguments passed to a function pointer*/

```
529     flag2++;
530 }
531 return ret;
```

01.w_Defects/wrong_arguments_func_pointer.c:539

Level Low

Status Not processed

```
536 int flag;
537 int (*fptr)(int,float);
538 fptr =wrong_arguments_func_pointer_017_func_002;

539 flag = fptr(1,4.5);

540 }
541
542 /*
```

Trace

fptr(1,4.5)

01.w_Defects/wrong_arguments_func_pointer.c:539

```
536 int flag;
537 int (*fptr)(int,float);
538 fptr =wrong_arguments_func_pointer_017_func_002;

539 flag = fptr(1,4.5);

540 }
541
542 /*
```

fptr(1,4.5)

01.w_Defects/wrong_arguments_func_pointer.c:539

```
536 int flag;
537 int (*fptr)(int,float);
538 fptr =wrong_arguments_func_pointer_017_func_002;

539 flag = fptr(1,4.5);

540 }
541
542 /*
```

01.w_Defects/zero_division.c:23

Level Low

Status Not processed

```
20 {
21 int dividend = 1000;
22 int ret;

23 ret = dividend / 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */
 */

24 }
25
26 /*
```

Trace

dividend / 0

01.w_Defects/zero_division.c:23

```
20 {
21 int dividend = 1000;
22 int ret;
```

```
23 ret = dividend / 0; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */
```

```
24 }  
25  
26 /*
```

```
dividend / 0
```

01.w_Defects/zero_division.c:23

```
20 {  
21     int dividend = 1000;  
22     int ret;
```

```
23     ret = dividend / 0; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */
```

```
24 }  
25  
26 /*
```

01.w_Defects/zero_division.c:35

Level Low

Status Not processed

```
32     int dividend = 1000;  
33     int ret;  
34     dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */
```

```
35     ret = dividend;
```

```
36 }  
37  
38 /*
```

Trace

ret = dividend

01.w_Defects/zero_division.c:35

```
32 int dividend = 1000;
33 int ret;
34 dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by
zero */

35 ret = dividend;

36 }
37
38 /*
```

ret = dividend

01.w_Defects/zero_division.c:35

```
32 int dividend = 1000;
33 int ret;
34 dividend /= 0; /*Tool should detect this line as error*/ /* ERROR:division by
zero */

35 ret = dividend;

36 }
37
38 /*
```

01.w_Defects/zero_division.c:47

Level Low

Status Not processed

```
44 {
45 int dividend = 1000;
46 int ret;
```

```
47 ret = dividend % 0; /*Tool should detect this line as error*/ /* ERROR:division by
zero */
```

```
48 }  
49  
50
```

Trace

dividend % 0

01.w_Defects/zero_division.c:47

```
44 {  
45   int dividend = 1000;  
46   int ret;
```

47 ret = dividend % 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */

```
48 }  
49  
50
```

dividend % 0

01.w_Defects/zero_division.c:47

```
44 {  
45   int dividend = 1000;  
46   int ret;
```

47 ret = dividend % 0; /*Tool should detect this line as error*/ /* ERROR:division by zero */

```
48 }  
49  
50
```

01.w_Defects/zero_division.c:78

Level Low

Status Not processed

```
75 int dividend = 1000;  
76 int divisors[5] = {2, 1, 0, 3, 4};  
77 int ret;
```

```
78 ret = dividend / divisors[2];/*Tool should detect this line as error*/ /* ERROR:division  
by zero */
```

```
79 }  
80  
81 /*
```

Trace

dividend / divisors[2]

01.w_Defects/zero_division.c:78

```
75 int dividend = 1000;  
76 int divisors[5] = {2, 1, 0, 3, 4};  
77 int ret;
```

```
78 ret = dividend / divisors[2];/*Tool should detect this line as error*/ /* ERROR:  
division by zero */
```

```
79 }  
80  
81 /*
```

dividend / divisors[2]

01.w_Defects/zero_division.c:78

```
75 int dividend = 1000;  
76 int divisors[5] = {2, 1, 0, 3, 4};  
77 int ret;
```

```
78 ret = dividend / divisors[2];/*Tool should detect this line as error*/ /* ERROR:  
division by zero */
```

```
79 }  
80  
81 /*
```

01.w_Defects/zero_division.c:93

Level Low

Status Not processed

```
90 int *p;  
91 int ret;  
92 p = &zero_division_006_gbl_divisor;  
  
93 ret = dividend / *p; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
94 }  
95  
96 /*
```

Trace

dividend / *p

01.w_Defects/zero_division.c:93

```
90 int *p;  
91 int ret;  
92 p = &zero_division_006_gbl_divisor;  
  
93 ret = dividend / *p; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
94 }  
95  
96 /*
```

dividend / *p

01.w_Defects/zero_division.c:93

```
90 int *p;
91 int ret;
92 p = &zero_division_006_gbl_divisor;

93 ret = dividend / *p; /*Tool should detect this line as error*/ /* ERROR:division
by zero */

94 }
95
96 /*
```

01.w_Defects/zero_division.c:118

Level Low

Status Not processed

```
115 int dividend = 1000;
116 int ret;
117 zero_division_007_func_001();

118 ret = dividend / zero_division_007_s_gbl.divisor; /*Tool should detect this line as
error*/ /* ERROR:division by zero */

119 }
120
121 /*
```

Trace

dividend / zero_division_007_s_gbl.divisor

01.w_Defects/zero_division.c:118

```
115 int dividend = 1000;
116 int ret;
117 zero_division_007_func_001();
```

```
118 ret = dividend / zero_division_007_s_gbl.divisor; /*Tool should detect this line  
as error*/ /* ERROR:division by zero */  
  
119 }  
120  
121 /*
```

dividend / zero_division_007_s_gbl.divisor

01.w_Defects/zero_division.c:118

```
115 int dividend = 1000;  
116 int ret;  
117 zero_division_007_func_001();  
  
118 ret = dividend / zero_division_007_s_gbl.divisor; /*Tool should detect this line  
as error*/ /* ERROR:division by zero */  
  
119 }  
120  
121 /*
```

01.w_Defects/zero_division.c:129

Level Low

Status Not processed

```
126 {  
127 float dividend = 1000.0;  
128 float ret;  
  
129 ret = dividend / 0.0; /*Tool should detect this line as error*/ /* ERROR:division by  
zero */  
  
130 }  
131  
132 /*
```

Trace

dividend / 0.0

01.w_Defects/zero_division.c:129

```
126 {  
127   float dividend = 1000.0;  
128   float ret;  
  
129   ret = dividend / 0.0; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */  
  
130 }  
131  
132 /*
```

dividend / 0.0

01.w_Defects/zero_division.c:129

```
126 {  
127   float dividend = 1000.0;  
128   float ret;  
  
129   ret = dividend / 0.0; /*Tool should detect this line as error*/ /* ERROR:division  
by zero */  
  
130 }  
131  
132 /*
```

01.w_Defects/zero_division.c:141

Level Low

Status Not processed

```
138 int dividend = 1000;  
139 int divisor = 0;  
140 int ret;
```

```
141 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:division by  
zero */
```

```
142 }  
143  
144 /*
```

Trace

dividend / divisor

01.w_Defects/zero_division.c:141

```
138 int dividend = 1000;  
139 int divisor = 0;  
140 int ret;
```

141 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:
division by zero */

```
142 }  
143  
144 /*
```

dividend / divisor

01.w_Defects/zero_division.c:141

```
138 int dividend = 1000;  
139 int divisor = 0;  
140 int ret;
```

141 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:
division by zero */

```
142 }  
143  
144 /*
```

01.w_Defects/zero_division.c:154

Level Low

Status Not processed

```
151 int divisor;
152 int ret;
153 divisor = rand();  
  
154 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:division by zero */  
  
155 }
156
157 /*
```

Trace

dividend / divisor

01.w_Defects/zero_division.c:154

```
151 int divisor;
152 int ret;
153 divisor = rand();  
  
154 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR: division by zero */  
  
155 }
156
157 /*
```

dividend / divisor

01.w_Defects/zero_division.c:154

```
151 int divisor;
152 int ret;
153 divisor = rand();  
  
154 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR: division by zero */
```

```
155 }  
156  
157 /*
```

01.w_Defects/zero_division.c:166

Level Low

Status Not processed

```
163 int dividend = 1000;  
164 int divisor = 2;  
165 int ret;
```

166 ret = dividend / (2 * divisor - 4);/*Tool should detect this line as error*/ /* ERROR:
division by zero */

```
167 }  
168  
169 /*
```

Trace

```
dividend / (2 * divisor - 4)
```

01.w_Defects/zero_division.c:166

```
163 int dividend = 1000;  
164 int divisor = 2;  
165 int ret;
```

166 ret = dividend / (2 * divisor - 4);/*Tool should detect this line as error*/ /*
ERROR:division by zero */

```
167 }  
168  
169 /*
```

```
dividend / (2 * divisor - 4)
```

01.w_Defects/zero_division.c:166

```
163 int dividend = 1000;  
164 int divisor = 2;  
165 int ret;  
  
166 ret = dividend / (2 * divisor - 4);/*Tool should detect this line as error*/ /*  
ERROR:division by zero */  
  
167 }  
168  
169 /*
```

01.w_Defects/zero_division.c:178

Level Low

Status Not processed

```
175 int dividend = 1000;  
176 int divisor = 2;  
177 int ret;
```

```
178 ret = dividend / (divisor * divisor - 4);/*Tool should detect this line as error*/ /*  
ERROR:division by zero */
```

```
179  
180 }  
181
```

Trace

```
dividend / (divisor * divisor - 4)
```

01.w_Defects/zero_division.c:178

```
175 int dividend = 1000;  
176 int divisor = 2;  
177 int ret;
```

```
178 ret = dividend / (divisor * divisor - 4);/*Tool should detect this line as error*/ /*  
ERROR:division by zero */
```

```
179  
180 }  
181
```

```
dividend / (divisor * divisor - 4)
```

01.w_Defects/zero_division.c:178

```
175 int dividend = 1000;  
176 int divisor = 2;  
177 int ret;
```

```
178 ret = dividend / (divisor * divisor - 4);/*Tool should detect this line as error*/ /*  
ERROR:division by zero */
```

```
179  
180 }  
181
```

01.w_Defects/zero_division.c:195

Level Low

Status Not processed

```
192 {  
193 int dividend = 1000;  
194 int ret;
```

```
195 ret = dividend / zero_division_013_func_001();/*Tool should detect this line as  
error*/ /* ERROR:division by zero */
```

```
196 }  
197  
198 /*
```

Trace

dividend / zero_division_013_func_001()

01.w_Defects/zero_division.c:195

```
192 {  
193 int dividend = 1000;  
194 int ret;  
  
195 ret = dividend / zero_division_013_func_001();/*Tool should detect this line  
as error*/ /* ERROR:division by zero */  
  
196 }  
197  
198 /*
```

dividend / zero_division_013_func_001()

01.w_Defects/zero_division.c:195

```
192 {  
193 int dividend = 1000;  
194 int ret;  
  
195 ret = dividend / zero_division_013_func_001();/*Tool should detect this line  
as error*/ /* ERROR:division by zero */  
  
196 }  
197  
198 /*
```

01.w_Defects/zero_division.c:206

Level Low**Status** Not processed

```
203 {  
204 int dividend = 1000;  
205 int ret;
```

```
206 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:division by  
zero */
```

```
207 }  
208  
209 void zero_division_014 ()
```

Trace

dividend / divisor

01.w_Defects/zero_division.c:206

```
203 {  
204 int dividend = 1000;  
205 int ret;
```

206 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:
division by zero */

```
207 }  
208  
209 void zero_division_014 ()
```

dividend / divisor

01.w_Defects/zero_division.c:206

```
203 {  
204 int dividend = 1000;  
205 int ret;
```

206 ret = dividend / divisor; /*Tool should detect this line as error*/ /* ERROR:
division by zero */

```
207 }  
208  
209 void zero_division_014 ()
```

01.w_Defects/zero_division.c:225

Level Low

Status Not processed

```
222 int divisor1;
223 int ret;
224 divisor1 = divisor;
```

```
225 ret = dividend / divisor1; /*Tool should detect this line as error*/ /* ERROR:division
by zero */
```

```
226 }
227
228 /*
```

Trace

```
dividend / divisor1
```

01.w_Defects/zero_division.c:225

```
222 int divisor1;
223 int ret;
224 divisor1 = divisor;
```

```
225 ret = dividend / divisor1; /*Tool should detect this line as error*/ /* ERROR:
division by zero */
```

```
226 }
227
228 /*
```

```
dividend / divisor1
```

01.w_Defects/zero_division.c:225

```
222 int divisor1;
223 int ret;
224 divisor1 = divisor;
```

```
225 ret = dividend / divisor1; /*Tool should detect this line as error*/ /* ERROR:
division by zero */
```

```
226 }  
227  
228 /*
```

01.w_Defects/zero_division.c:252

Level Low

Status Not processed

```
249 zero_division_016_func_002 ();  
250 divisor1 = *zero_division_016_gbl_divisor;  
251 divisor2 = divisor1;
```

252 ret = dividend / divisor2; /*Tool should detect this line as error*/ /* ERROR:division by zero */

```
253 }  
254  
255 /*
```

Trace

dividend / divisor2

01.w_Defects/zero_division.c:252

```
249 zero_division_016_func_002 ();  
250 divisor1 = *zero_division_016_gbl_divisor;  
251 divisor2 = divisor1;
```

252 ret = dividend / divisor2; /*Tool should detect this line as error*/ /* ERROR:division by zero */

```
253 }  
254  
255 /*
```

dividend / divisor2

01.w_Defects/zero_division.c:252

```
249 zero_division_016_func_002();  
250 divisor1 = *zero_division_016_gbl_divisor;  
251 divisor2 = divisor1;  
  
252 ret = dividend / divisor2; /*Tool should detect this line as error*/ /* ERROR:  
division by zero */  
  
253 }  
254  
255 /*
```

02.wo_Defects/free_nondynamically_allocated_memory.c:60

Level Low**Status** Not processed

```
57 void free_nondynamic_allocated_memory_004()  
58 {  
59     char* ptr1="a";  
  
60     char** ptr=&ptr1;  
  
61     while(0)  
62     free(ptr); /*Tool should not detect this line as error*/ /*No ERROR:Free memory not  
allocated dynamically*/  
63 }
```

Trace

&ptr1

02.wo_Defects/free_nondynamically_allocated_memory.c:60

```
57 void free_nondynamic_allocated_memory_004()  
58 {  
59     char* ptr1="a";
```

```
60 char** ptr=&ptr1;

61 while(0)
62 free(ptr); /*Tool should not detect this line as error*/ /*No ERROR:Free
memory not allocated dynamically*/
63 }
```

&ptr1

02.wo_Defects/free_nondynamically_allocated_memory.c:60

```
57 void free_nondynamic_allocated_memory_004()
58 {
59     char* ptr1="a";

60     char** ptr=&ptr1;

61     while(0)
62     free(ptr); /*Tool should not detect this line as error*/ /*No ERROR:Free
memory not allocated dynamically*/
63 }
```

02.wo_Defects/free_null_pointer.c:364

Level Low**Status** Not processed

```
361 static free_null_pointer_011_u_001 *u;
362 free_null_pointer_011_u_001 * free_null_pointer_011_func_001 ()
363 {

364     int flag = rand();

365     flag = 1;
366     switch (flag)
367 {
```

Trace

rand()

02.wo_Defects/free_null_pointer.c:364

```
361 static free_null_pointer_011_u_001 *u;
362 free_null_pointer_011_u_001 * free_null_pointer_011_func_001 ()
363 {

364 int flag = rand();

365 flag = 1;
366 switch (flag)
367 {
```

rand()

02.wo_Defects/free_null_pointer.c:364

```
361 static free_null_pointer_011_u_001 *u;
362 free_null_pointer_011_u_001 * free_null_pointer_011_func_001 ()
363 {

364 int flag = rand();

365 flag = 1;
366 switch (flag)
367 {
```

02.wo_Defects/free_null_pointer.c:425

Level Low**Status** Not processed

```
422 free_null_pointer_011_u_001 *p;
423 p = free_null_pointer_011_func_001 ();
424 ret = p->b;

425 p = free_null_pointer_011_func_002 ();

426 sink = ret;
```

```
427 }  
428
```

Trace

free_null_pointer_011_func_002 ()

02.wo_Defects/free_null_pointer.c:425

```
422 free_null_pointer_011_u_001 *p;  
423 p = free_null_pointer_011_func_001 ();  
424 ret = p->b;  
  
425 p = free_null_pointer_011_func_002 ();  
  
426     sink = ret;  
427 }  
428
```

free_null_pointer_011_func_002 ()

02.wo_Defects/free_null_pointer.c:425

```
422 free_null_pointer_011_u_001 *p;  
423 p = free_null_pointer_011_func_001 ();  
424 ret = p->b;  
  
425 p = free_null_pointer_011_func_002 ();  
  
426     sink = ret;  
427 }  
428
```

02.wo_Defects/free_null_pointer.c:458

Level Low

Status Not processed

455 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)

```
456 {  
457   if(flag == 10)  
  
458   a = *(ptr+1);  
  
459 }  
460  
461 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
```

Trace

```
*(ptr+1)
```

02.wo_Defects/free_null_pointer.c:458

```
455 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)  
456 {  
457   if(flag == 10)  
  
458   a = *(ptr+1);  
  
459 }  
460  
461 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
```

```
*(ptr+1)
```

02.wo_Defects/free_null_pointer.c:458

```
455 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)  
456 {  
457   if(flag == 10)  
  
458   a = *(ptr+1);  
  
459 }  
460  
461 if (free_null_pointer_012_func_001(0) == ZERO && MAX ==1)
```

02.wo_Defects/func_pointer.c:341

Level Low**Status** Not processed

```
338
339 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
340 {

341 int flag = rand();

342 flag = 1;
343 func_pointer_009_u_001 *u;
344 switch (flag)
```

Trace

rand()

02.wo_Defects/func_pointer.c:341

```
338
339 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
340 {

341 int flag = rand();

342 flag = 1;
343 func_pointer_009_u_001 *u;
344 switch (flag)
```

rand()

02.wo_Defects/func_pointer.c:341

```
338
339 func_pointer_009_u_001 * func_pointer_009_func_001 (void)
340 {

341 int flag = rand();

342 flag = 1;
```

```
343 func_pointer_009_u_001 *u;
344 switch (flag)
```

02.wo_Defects/func_pointer.c:513

Level Low**Status** Not processed

```
510 {
511     int (*func_gbl)(void );
512     func_gbl = func_pointer_013_func_001;

513     flag = func_gbl();/*Tool should not detect this line as error*/ /*No ERROR:Bad
function pointer casting*/

514 }
515 ptr = &arr[0];
516 *(ptr+1) = 7;
```

Trace

func_gbl()

02.wo_Defects/func_pointer.c:513

```
510 {
511     int (*func_gbl)(void );
512     func_gbl = func_pointer_013_func_001;

513     flag = func_gbl();/*Tool should not detect this line as error*/ /*No ERROR:
Bad function pointer casting*/

514 }
515 ptr = &arr[0];
516 *(ptr+1) = 7;
```

func_gbl()

02.wo_Defects/func_pointer.c:513

```
510 {  
511     int (*func_gbl)(void );  
512     func_gbl = func_pointer_013_func_001;  
  
513     flag = func_gbl();/*Tool should not detect this line as error*/ /*No ERROR:  
Bad function pointer casting*/  
  
514 }  
515 ptr = &arr[0];  
516 *(ptr+1) = 7;
```

02.wo_Defects/func_pointer.c:565

Level Low**Status** Not processed

```
562 my_label2:  
563     if (flag == 1)  
564     {  
  
565         flag = func_gbl();/*Tool should not detect this line as error*/ /*No ERROR:Bad  
function pointer casting*/  
  
566     }  
567     return ret;  
568 }
```

Trace

func_gbl()

02.wo_Defects/func_pointer.c:565

```
562 my_label2:  
563     if (flag == 1)  
564     {
```

```
565     flag = func_gbl();/*Tool should not detect this line as error*/ /*No  
ERROR:Bad function pointer casting*/
```

```
566 }  
567 return ret;  
568 }
```

func_gbl()

02.wo_Defects/func_pointer.c:565

```
562 my_label2:  
563     if (flag == 1)  
564     {
```

```
565     flag = func_gbl();/*Tool should not detect this line as error*/ /*No  
ERROR:Bad function pointer casting*/
```

```
566 }  
567 return ret;  
568 }
```

02.wo_Defects/littlemem_st.c:37

Level Low

Status Not processed

```
34 }  
35  
36 p = (littlemem_st_001_s_001 *)buf;
```

```
37 ret = p->c; /*Tool should not detect this line as error*/ /*No ERROR:Little Memory or  
Overflow*/
```

```
38 printf("%d \n",p->c);  
39 }  
40
```

Trace

p->c

02.wo_Defects/littlemem_st.c:37

```
34  }
35
36  p = (littlemem_st_001_s_001 *)buf;

37  ret = p->c; /*Tool should not detect this line as error*/ /*No ERROR:Little
Memory or Overflow*/

38  printf("%d \n",p->c);
39 }
40
```

p->c

02.wo_Defects/littlemem_st.c:37

```
34  }
35
36  p = (littlemem_st_001_s_001 *)buf;

37  ret = p->c; /*Tool should not detect this line as error*/ /*No ERROR:Little
Memory or Overflow*/

38  printf("%d \n",p->c);
39 }
40
```

02.wo_Defects/lock_never_unlock.c:231

Level Low**Status** Not processed

```
228
229  pthread_mutex_lock( &lock_never_unlock_004_glb_mutex_2 );
230  ip = (long)input;

231  ip = ip *20;
```

```
232
233 #if defined PRINT_DEBUG
234   printf("Task4_2! Lock Never Unlock, thread #%ld!\n",ip);
```

Trace

```
ip *20
```

02.wo_Defects/lock_never_unlock.c:231

```
228
229   pthread_mutex_lock( &lock_never_unlock_004_glb_mutex_2 );
230   ip = (long)input;
```

```
231   ip = ip *20;
```

```
232
```

```
233 #if defined PRINT_DEBUG
234   printf("Task4_2! Lock Never Unlock, thread #%ld!\n",ip);
```

```
ip *20
```

02.wo_Defects/lock_never_unlock.c:231

```
228
229   pthread_mutex_lock( &lock_never_unlock_004_glb_mutex_2 );
230   ip = (long)input;
```

```
231   ip = ip *20;
```

```
232
```

```
233 #if defined PRINT_DEBUG
234   printf("Task4_2! Lock Never Unlock, thread #%ld!\n",ip);
```

02.wo_Defects/lock_never_unlock.c:327

Level Low

Status Not processed

```
324
325 pthread_mutex_lock( &lock_never_unlock_006_glb_mutex_2 );
326 ip = (long)input;

327 ip = ip *20;

328
329 #if defined PRINT_DEBUG
330 printf("Task6_2! Lock Never Unlock, thread #%ld!\n",ip);
```

Trace

```
ip *20
```

02.wo_Defects/lock_never_unlock.c:327

```
324
325 pthread_mutex_lock( &lock_never_unlock_006_glb_mutex_2 );
326 ip = (long)input;

327 ip = ip *20;

328
329 #if defined PRINT_DEBUG
330 printf("Task6_2! Lock Never Unlock, thread #%ld!\n",ip);
```

```
ip *20
```

02.wo_Defects/lock_never_unlock.c:327

```
324
325 pthread_mutex_lock( &lock_never_unlock_006_glb_mutex_2 );
326 ip = (long)input;

327 ip = ip *20;

328
329 #if defined PRINT_DEBUG
330 printf("Task6_2! Lock Never Unlock, thread #%ld!\n",ip);
```

02.wo_Defects/memory_allocation_failure.c:292

Level Low**Status** Not processed

```
289     int i=0;
290     do
291     {
292         buf = (char*) malloc(MAX_BUFFER * sizeof(char)); /*Tool should not detect
this line as error*/ /*No ERROR:Memory allocation failure */
293         i++;
294     }while (i<MAX_VAL);
295 }
```

Trace

(char*) malloc(MAX_BUFFER * sizeof(char))

02.wo_Defects/memory_allocation_failure.c:292

```
289     int i=0;
290     do
291     {
292         buf = (char*) malloc(MAX_BUFFER * sizeof(char)); /*Tool should not
detect this line as error*/ /*No ERROR:Memory allocation failure */
293         i++;
294     }while (i<MAX_VAL);
295 }
```

(char*) malloc(MAX_BUFFER * sizeof(char))

02.wo_Defects/memory_allocation_failure.c:292

```
289     int i=0;
290     do
291     {
292         buf = (char*) malloc(MAX_BUFFER * sizeof(char)); /*Tool should not
```

detect this line as error*/ /*No ERROR:Memory allocation failure */

```
293         i++;
294     }while (i<MAX_VAL);
295 }
```

02.wo_Defects/memory_allocation_failure.c:457

Level Low

Status Not processed

```
454 if(flag == 10){
455   if(memory_allocation_failure_012_buf2_gbl!=NULL)
456   {
457       a = ptr[1][1];
458   free(memory_allocation_failure_012_buf2_gbl);
459 }
460 }
```

Trace

ptr[1][1]

02.wo_Defects/memory_allocation_failure.c:457

```
454 if(flag == 10){
455   if(memory_allocation_failure_012_buf2_gbl!=NULL)
456   {
457       a = ptr[1][1];
458   free(memory_allocation_failure_012_buf2_gbl);
459 }
460 }
```

ptr[1][1]

02.wo_Defects/memory_allocation_failure.c:457

```
454 if(flag == 10){  
455   if(memory_allocation_failure_012_buf2_gbl!=NULL)  
456   {  
  
457     a = ptr[1][1];  
  
458     free(memory_allocation_failure_012_buf2_gbl);  
459   }  
460 }
```

02.wo_Defects/memory_leak.c:378

Level Low**Status** Not processed

```
375 float **fp2 = &fptr;  
376 fptr = NULL;  
377 {
```

```
378   float * fptr = *fp1;
```

```
379   fptr = (float *)calloc(10, sizeof(float)); /*Tool should not detect this line as error*/  
/*No ERROR:Memory Leakage */  
380   if(fptr!=NULL)  
381   {
```

Trace

*fp1

02.wo_Defects/memory_leak.c:378

```
375 float **fp2 = &fptr;  
376 fptr = NULL;  
377 {
```

```
378 float * fptr = *fp1;
```

```
379 fptr = (float *)calloc(10, sizeof(float)); /*Tool should not detect this line as  
error*/ /*No ERROR:Memory Leakage */  
380 if(fptr!=NULL)  
381 {
```

*fp1

02.wo_Defects/memory_leak.c:378

```
375 float **fp2 = &fptr;  
376 fptr = NULL;  
377 {
```

```
378 float * fptr = *fp1;
```

```
379 fptr = (float *)calloc(10, sizeof(float)); /*Tool should not detect this line as  
error*/ /*No ERROR:Memory Leakage */  
380 if(fptr!=NULL)  
381 {
```

02.wo_Defects/race_condition.c:222

Level Low

Status Not processed

```
219 race_condition_005_glb_data = (race_condition_005_glb_data % 100) + 1; /*Tool  
should not detect this line as error*/ /*No ERROR:Race condition*/  
220 pthread_mutex_unlock(&race_condition_005_glb_mutex);  
221
```

```
222 unsigned long ip = (unsigned long)pthread_self();
```

```
223 // printf("Task1! Lock Never Unlock, threadID# %lu! gbl1 = %d \n",ip ,  
race_condition_005_glb_data);  
224 #endif /* defined(CHECKER_POLYSPACE) */  
225 return NULL;
```

Trace

(unsigned long)pthread_self()

02.wo_Defects/race_condition.c:222

```
219 race_condition_005_glb_data = (race_condition_005_glb_data % 100) + 1;  
/*Tool should not detect this line as error*/ /*No ERROR:Race condition*/  
220 pthread_mutex_unlock(&race_condition_005_glb_mutex);  
221  
  
222 unsigned long ip = (unsigned long)pthread_self();  
  
223 // printf("Task1! Lock Never Unlock, threadID# %lu! gbl1 = %d \n",ip ,  
race_condition_005_glb_data);  
224 #endif /* defined(CHECKER_POLYSPACE) */  
225 return NULL;
```

(unsigned long)pthread_self()

02.wo_Defects/race_condition.c:222

```
219 race_condition_005_glb_data = (race_condition_005_glb_data % 100) + 1;  
/*Tool should not detect this line as error*/ /*No ERROR:Race condition*/  
220 pthread_mutex_unlock(&race_condition_005_glb_mutex);  
221  
  
222 unsigned long ip = (unsigned long)pthread_self();  
  
223 // printf("Task1! Lock Never Unlock, threadID# %lu! gbl1 = %d \n",ip ,  
race_condition_005_glb_data);  
224 #endif /* defined(CHECKER_POLYSPACE) */  
225 return NULL;
```

02.wo_Defects/race_condition.c:420

Level Low

Status Not processed

```
417  
418 pthread_mutex_lock( &race_condition_009_glb_mutex_1 );  
419 ip = (long)input;
```

```
420 ip = ip *10;

421 race_condition_009_glb_data++; /*Tool should not detect this line as error*/ /*No
ERROR:Race condition*/
422 #if defined PRINT_DEBUG
423 printf("Task4_1! Lock Never Unlock, thread #%ld!\n",ip);
```

Trace

```
ip *10
```

02.wo_Defects/race_condition.c:420

```
417
418 pthread_mutex_lock( &race_condition_009_glb_mutex_1 );
419 ip = (long)input;
```

```
420 ip = ip *10;
```

```
421 race_condition_009_glb_data++; /*Tool should not detect this line as error*/
/*No ERROR:Race condition*/
422 #if defined PRINT_DEBUG
423 printf("Task4_1! Lock Never Unlock, thread #%ld!\n",ip);
```

```
ip *10
```

02.wo_Defects/race_condition.c:420

```
417
418 pthread_mutex_lock( &race_condition_009_glb_mutex_1 );
419 ip = (long)input;
```

```
420 ip = ip *10;
```

```
421 race_condition_009_glb_data++; /*Tool should not detect this line as error*/
/*No ERROR:Race condition*/
422 #if defined PRINT_DEBUG
423 printf("Task4_1! Lock Never Unlock, thread #%ld!\n",ip);
```

02.wo_Defects/race_condition.c:440

Level Low**Status** Not processed

```
437
438 pthread_mutex_lock( &race_condition_009_glb_mutex_2 );
439 ip = (long)input;

440 ip = ip *20;

441 race_condition_009_glb_data--;
442 #if defined PRINT_DEBUG
443 printf("Task4_2! Lock Never Unlock, thread #%ld!\n",ip);
```

Trace

ip *20

02.wo_Defects/race_condition.c:440

```
437
438 pthread_mutex_lock( &race_condition_009_glb_mutex_2 );
439 ip = (long)input;

440 ip = ip *20;

441 race_condition_009_glb_data--;
442 #if defined PRINT_DEBUG
443 printf("Task4_2! Lock Never Unlock, thread #%ld!\n",ip);
```

ip *20

02.wo_Defects/race_condition.c:440

```
437
438 pthread_mutex_lock( &race_condition_009_glb_mutex_2 );
439 ip = (long)input;

440 ip = ip *20;

441 race_condition_009_glb_data--;
```

```
442 #if defined PRINT_DEBUG  
443     printf("Task4_2! Lock Never Unlock, thread #%ld!\n",ip);
```

02.wo_Defects/uninit_memory_access.c:100

Level Low**Status** Not processed

```
97 }  
98 }  
99 }
```

100 k = arr1[1][2][3]; /*Tool should not detect this line as error*/ /*No ERROR:
Uninitialized Memory Access*/

```
101 }  
102  
103 /*
```

Trace

arr1[1][2][3]

02.wo_Defects/uninit_memory_access.c:100

```
97 }  
98 }  
99 }
```

100 k = arr1[1][2][3]; /*Tool should not detect this line as error*/ /*No ERROR:
Uninitialized Memory Access*/

```
101 }  
102  
103 /*
```

```
arr1[1][2][3]
```

02.wo_Defects/uninit_memory_access.c:100

```
97 }  
98 }  
99 }
```

100 k = arr1[1][2][3]; /*Tool should not detect this line as error*/ /*No ERROR:
Uninitialized Memory Access*/

```
101 }  
102 /*  
103 *
```

02.wo_Defects/uninit_memory_access.c:401

Level Low

Status Not processed

```
398  
399 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001 ()  
400 {
```

401 int flag = rand();

```
402 flag = 1;  
403 uninit_memory_access_014_u_001 *u;  
404 switch (flag)
```

Trace

```
rand()
```

02.wo_Defects/uninit_memory_access.c:401

```
398  
399 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001  
()  
400 {
```

```
401 int flag = rand();  
  
402 flag = 1;  
403 uninit_memory_access_014_u_001 *u;  
404 switch (flag)
```

rand()

02.wo_Defects/uninit_memory_access.c:401

```
398  
399 uninit_memory_access_014_u_001 * uninit_memory_access_014_func_001  
(  
400 {  
  
401 int flag = rand();  
  
402 flag = 1;  
403 uninit_memory_access_014_u_001 *u;  
404 switch (flag)
```

02.wo_Defects/wrong_arguments_func_pointer.c:53

Level Low

Status Not processed

```
50 int (*fptr)(int);  
51 int a;  
52 fptr = wrong_arguments_func_pointer_001_func_001;  
  
53 a =fptr(arr[0]); /*Tool should not detect this line as error**/*No ERROR:Wrong  
arguments passed to a function pointer*/  
  
54 }  
55  
56 /*
```

Trace

fptr(arr[0])

02.wo_Defects/wrong_arguments_func_pointer.c:53

```
50 int (*fptr)(int);
51 int a;
52 fptr = wrong_arguments_func_pointer_001_func_001;

53 a =fptr(arr[0]); /*Tool should not detect this line as error__*//*No ERROR:
Wrong arguments passed to a function pointer*/

54 }
55
56 /*
```

fptr(arr[0])

02.wo_Defects/wrong_arguments_func_pointer.c:53

```
50 int (*fptr)(int);
51 int a;
52 fptr = wrong_arguments_func_pointer_001_func_001;

53 a =fptr(arr[0]); /*Tool should not detect this line as error__*//*No ERROR:
Wrong arguments passed to a function pointer*/

54 }
55
56 /*
```

02.wo_Defects/wrong_arguments_func_pointer.c:74

Level Low

Status Not processed

```
71 int (*fptr)(int);
72 int a;
73 fptr = wrong_arguments_func_pointer_002_func_001;
```

```
74 a =fptr(buf[0]); /*Tool should not detect this line as error__*//*No ERROR:Wrong
arguments passed to a function pointer*/
```

```
75 }  
76  
77 /*
```

Trace

fptr(buf[0])

02.wo_Defects/wrong_arguments_func_pointer.c:74

```
71 int (*fptr)(int);  
72 int a;  
73 fptr = wrong_arguments_func_pointer_002_func_001;
```

74 a =fptr(buf[0]); /*Tool should not detect this line as error**/*No ERROR:
Wrong arguments passed to a function pointer*/

```
75 }  
76  
77 /*
```

fptr(buf[0])

02.wo_Defects/wrong_arguments_func_pointer.c:74

```
71 int (*fptr)(int);  
72 int a;  
73 fptr = wrong_arguments_func_pointer_002_func_001;
```

74 a =fptr(buf[0]); /*Tool should not detect this line as error**/*No ERROR:
Wrong arguments passed to a function pointer*/

```
75 }  
76  
77 /*
```

02.wo_Defects/wrong_arguments_func_pointer.c:94

Level Low

Status Not processed

```
91 int a = 1;
92 int ret;
93 func = wrong_arguments_func_pointer_003_func_001;

94 ret = func(&a); /*Tool should not detect this line as error__*//*No ERROR:Wrong
arguments passed to a function pointer*/

95 }
96
97 /*
```

Trace

func(&a)

02.wo_Defects/wrong_arguments_func_pointer.c:94

```
91 int a = 1;
92 int ret;
93 func = wrong_arguments_func_pointer_003_func_001;

94 ret = func(&a); /*Tool should not detect this line as error__*//*No ERROR:
Wrong arguments passed to a function pointer*/

95 }
96
97 /*
```

func(&a)

02.wo_Defects/wrong_arguments_func_pointer.c:94

```
91 int a = 1;
92 int ret;
93 func = wrong_arguments_func_pointer_003_func_001;

94 ret = func(&a); /*Tool should not detect this line as error__*//*No ERROR:
Wrong arguments passed to a function pointer*/
```

```
95 }  
96  
97 /*
```

02.wo_Defects/wrong_arguments_func_pointer.c:114

Level Low

Status Not processed

```
111 char buf[10] = "string";  
112 char ret;  
113 func = wrong_arguments_func_pointer_004_func_001;  
  
114 ret = func(buf); /*Tool should not detect this line as error**/*No ERROR:Wrong  
arguments passed to a function pointer*/  
  
115 }  
116  
117 /*
```

Trace

func(buf)

02.wo_Defects/wrong_arguments_func_pointer.c:114

```
111 char buf[10] = "string";  
112 char ret;  
113 func = wrong_arguments_func_pointer_004_func_001;  
  
114 ret = func(buf); /*Tool should not detect this line as error**/*No ERROR:  
Wrong arguments passed to a function pointer*/  
  
115 }  
116  
117 /*
```

```
func(buf)
```

02.wo_Defects/wrong_arguments_func_pointer.c:114

```
111 char buf[10] = "string";
112 char ret;
113 func = wrong_arguments_func_pointer_004_func_001;

114 ret = func(buf); /*Tool should not detect this line as error//No ERROR:
Wrong arguments passed to a function pointer*/

115 }
116
117 /*
```

02.wo_Defects/wrong_arguments_func_pointer.c:139

Level Low

Status Not processed

```
136 char buf[10] = "string";
137 float ret;
138 func = wrong_arguments_func_pointer_005_func_001;

139 ret = func(buf); /*Tool should not detect this line as error//No ERROR:Wrong
arguments passed to a function pointer*/

140 }
141
142 /*
```

Trace

```
func(buf)
```

02.wo_Defects/wrong_arguments_func_pointer.c:139

```
136 char buf[10] = "string";
137 float ret;
138 func = wrong_arguments_func_pointer_005_func_001;
```

```
139 ret = func(buf); /*Tool should not detect this line as error__*//*No ERROR:  
Wrong arguments passed to a function pointer*/
```

```
140 }  
141  
142 /*
```

```
func(buf)
```

02.wo_Defects/wrong_arguments_func_pointer.c:139

```
136 char buf[10] = "string";  
137 float ret;  
138 func = wrong_arguments_func_pointer_005_func_001;
```

```
139 ret = func(buf); /*Tool should not detect this line as error__*//*No ERROR:  
Wrong arguments passed to a function pointer*/
```

```
140 }  
141  
142 /*
```

02.wo_Defects/wrong_arguments_func_pointer.c:158

Level Low

Status Not processed

```
155 int (*func)(int, int);  
156 int ret;  
157 func = wrong_arguments_func_pointer_006_func_001;
```

```
158 ret = func(1, 2); /*Tool should not detect this line as error__*//*No ERROR:Wrong  
arguments passed to a function pointer*/
```

```
159 }  
160  
161 /*
```

Trace

```
func(1, 2)
```

02.wo_Defects/wrong_arguments_func_pointer.c:158

```
155 int (*func)(int, int);
156 int ret;
157 func = wrong_arguments_func_pointer_006_func_001;

158 ret = func(1, 2); /*Tool should not detect this line as error**/*No ERROR:
Wrong arguments passed to a function pointer*/

159 }
160
161 /*
```

```
func(1, 2)
```

02.wo_Defects/wrong_arguments_func_pointer.c:158

```
155 int (*func)(int, int);
156 int ret;
157 func = wrong_arguments_func_pointer_006_func_001;

158 ret = func(1, 2); /*Tool should not detect this line as error**/*No ERROR:
Wrong arguments passed to a function pointer*/

159 }
160
161 /*
```

02.wo_Defects/wrong_arguments_func_pointer.c:178

Level Low

Status Not processed

```
175 unsigned int (*func)(double);
176 unsigned int ret;
177 func = wrong_arguments_func_pointer_007_func_001;
```

```
178 ret = func(10.005); /*Tool should not detect this line as error**/*No ERROR:Wrong
arguments passed to a function pointer*/
```

```
179 }  
180  
181 /*
```

Trace

```
func(10.005)
```

02.wo_Defects/wrong_arguments_func_pointer.c:178

```
175 unsigned int (*func)(double);  
176 unsigned int ret;  
177 func = wrong_arguments_func_pointer_007_func_001;  
  
178 ret = func(10.005); /*Tool should not detect this line as error**/*No ERROR:  
Wrong arguments passed to a function pointer*/
```

```
179 }  
180  
181 /*
```

```
func(10.005)
```

02.wo_Defects/wrong_arguments_func_pointer.c:178

```
175 unsigned int (*func)(double);  
176 unsigned int ret;  
177 func = wrong_arguments_func_pointer_007_func_001;  
  
178 ret = func(10.005); /*Tool should not detect this line as error**/*No ERROR:  
Wrong arguments passed to a function pointer*/
```

```
179 }  
180  
181 /*
```

02.wo_Defects/wrong_arguments_func_pointer.c:197

Level Low

Status Not processed

```
194 float a = 11.5;
195 float ret;
196 func_glb = wrong_arguments_func_pointer_008_func_001;

197 ret = func_glb(a); /*Tool should not detect this line as error__*//*No ERROR:Wrong
arguments passed to a function pointer*/

198 }
199
200 /*
```

Trace

func_glb(a)

02.wo_Defects/wrong_arguments_func_pointer.c:197

```
194 float a = 11.5;
195 float ret;
196 func_glb = wrong_arguments_func_pointer_008_func_001;

197 ret = func_glb(a); /*Tool should not detect this line as error__*//*No ERROR:
Wrong arguments passed to a function pointer*/

198 }
199
200 /*
```

func_glb(a)

02.wo_Defects/wrong_arguments_func_pointer.c:197

```
194 float a = 11.5;
195 float ret;
196 func_glb = wrong_arguments_func_pointer_008_func_001;

197 ret = func_glb(a); /*Tool should not detect this line as error__*//*No ERROR:
Wrong arguments passed to a function pointer*/
```

```
198 }  
199  
200 /*
```

02.wo_Defects/wrong_arguments_func_pointer.c:224

Level Low

Status Not processed

```
221 char ret;  
222 char (*func)(char *,char *, char *);  
223 func = wrong_arguments_func_pointer_009_func_001;
```

224 ret = func(str1,str2,str3); /*Tool should not detect this line as error**/*No ERROR:
Wrong arguments passed to a function pointer*/

```
225 free(str1);  
226 free(str2);  
227 free(str3);
```

Trace

```
func(str1,str2,str3)
```

02.wo_Defects/wrong_arguments_func_pointer.c:224

```
221 char ret;  
222 char (*func)(char *,char *, char *);  
223 func = wrong_arguments_func_pointer_009_func_001;
```

224 ret = func(str1,str2,str3); /*Tool should not detect this line as error**/*No
ERROR:Wrong arguments passed to a function pointer*/

```
225 free(str1);  
226 free(str2);  
227 free(str3);
```

```
func(str1,str2,str3)
```

02.wo_Defects/wrong_arguments_func_pointer.c:224

```
221 char ret;
222 char (*func)(char *,char *, char *);
223 func = wrong_arguments_func_pointer_009_func_001;
```

```
224 ret = func(str1,str2,str3); /*Tool should not detect this line as error**/*No
ERROR:Wrong arguments passed to a function pointer*/
```

```
225 free(str1);
226 free(str2);
227 free(str3);
```

02.wo_Defects/wrong_arguments_func_pointer.c:284

Level Low

Status Not processed

```
281 for (i = 0; i < MAX; i++)
282 {
283     st->arr[i] = i;
284     temp = st->arr[i];
285 }
286 }
287
```

Trace

```
st->arr[i]
```

02.wo_Defects/wrong_arguments_func_pointer.c:284

```
281 for (i = 0; i < MAX; i++)
282 {
283     st->arr[i] = i;
```

```
284     temp = st->arr[i];
```

```
285 }
286 }
287
```

st->arr[i]

02.wo_Defects/wrong_arguments_func_pointer.c:284

```
281 for (i = 0; i < MAX; i++)
282 {
283     st->arr[i] = i;
```

```
284     temp = st->arr[i];
```

```
285 }
286 }
287
```

02.wo_Defects/wrong_arguments_func_pointer.c:420

Level Low

Status Not processed

```
417 long (*fptr)(long [],int);
418 long a;
419 fptr = wrong_arguments_func_pointer_014_func_002;
```

```
420 a =fptr(arr,5); /*Tool should not detect this line as error**/*No ERROR:Wrong
arguments passed to a function pointer*/
```

```
421 }
422 }
423
```

Trace

fptr(arr,5)

02.wo_Defects/wrong_arguments_func_pointer.c:420

```
417     long (*fptr)(long [],int);
418     long a;
419     fptr = wrong_arguments_func_pointer_014_func_002;

420     a =fptr(arr,5); /*Tool should not detect this line as error//No ERROR:
Wrong arguments passed to a function pointer*/

421 }
422 }
423
```

fptr(arr,5)

02.wo_Defects/wrong_arguments_func_pointer.c:420

```
417     long (*fptr)(long [],int);
418     long a;
419     fptr = wrong_arguments_func_pointer_014_func_002;

420     a =fptr(arr,5); /*Tool should not detect this line as error//No ERROR:
Wrong arguments passed to a function pointer*/

421 }
422 }
423
```

02.wo_Defects/wrong_arguments_func_pointer.c:487

Level Low

Status Not processed

```
484 char ret;
485 char (*func)(char *,int *, float *);
486 func = wrong_arguments_func_pointer_016_func_001; /*Tool should not detect this
line as error//No ERROR:Wrong arguments passed to a function pointer*/

487 ret = func(str1,str2,str3);
```

```
488 }  
489  
490 /*
```

Trace

```
func(str1,str2,str3)
```

02.wo_Defects/wrong_arguments_func_pointer.c:487

```
484 char ret;  
485 char (*func)(char *,int *, float *);  
486 func = wrong_arguments_func_pointer_016_func_001; /*Tool should not  
detect this line as error**/*No ERROR:Wrong arguments passed to a function  
pointer*/  
  
487 ret = func(str1,str2,str3);
```

```
488 }  
489  
490 /*
```

```
func(str1,str2,str3)
```

02.wo_Defects/wrong_arguments_func_pointer.c:487

```
484 char ret;  
485 char (*func)(char *,int *, float *);  
486 func = wrong_arguments_func_pointer_016_func_001; /*Tool should not  
detect this line as error**/*No ERROR:Wrong arguments passed to a function  
pointer*/  
  
487 ret = func(str1,str2,str3);
```

```
488 }  
489  
490 /*
```

02.wo_Defects/wrong_arguments_func_pointer.c:498

Level Low**Status** Not processed

```
495 int wrong_arguments_func_pointer_017_func_001 (int flag,float flag2)
496 {
497     float a=0.0;

498     a += flag2;

499     flag = 1;
500     return flag;
501 }
```

Trace

a += flag2

02.wo_Defects/wrong_arguments_func_pointer.c:498

```
495 int wrong_arguments_func_pointer_017_func_001 (int flag,float flag2)
496 {
497     float a=0.0;

498     a += flag2;

499     flag = 1;
500     return flag;
501 }
```

a += flag2

02.wo_Defects/wrong_arguments_func_pointer.c:498

```
495 int wrong_arguments_func_pointer_017_func_001 (int flag,float flag2)
496 {
497     float a=0.0;

498     a += flag2;

499     flag = 1;
```

```
500 return flag;
501 }
```

02.wo_Defects/wrong_arguments_func_pointer.c:525

Level Low

Status Not processed

```
522 my_label2:
523   if (flag == 1)
524 {
```

525 flag = wrong_arguments_func_pointer_017_func_gbl(0,1.9); /*Tool should not detect this line as error**/*No ERROR:Wrong arguments passed to a function pointer*/

```
526     flag2++;
527 }
528 return ret;
```

Trace

```
wrong_arguments_func_pointer_017_func_
gbl(0,1.9)
```

02.wo_Defects/wrong_arguments_func_pointer.c:525

```
522 my_label2:
523   if (flag == 1)
524 {
```

525 flag = wrong_arguments_func_pointer_017_func_gbl(0,1.9); /*Tool should not detect this line as error**/*No ERROR:Wrong arguments passed to a function pointer*/

```
526     flag2++;
527 }
528 return ret;
```

wrong_arguments_func_pointer_017_func_gbl(0,1.9)

02.wo_Defects/wrong_arguments_func_pointer.c:525

```
522 my_label2:  
523     if (flag == 1)  
524     {  
  
525         flag = wrong_arguments_func_pointer_017_func_gbl(0,1.9); /*Tool  
should not detect this line as error*//*No ERROR:Wrong arguments passed to a  
function pointer*/  
  
526         flag2++;  
527     }  
528     return ret;
```

02.wo_Defects/wrong_arguments_func_pointer.c:536

Level Low**Status** Not processed

```
533 int flag;  
534 int (*fptr)(int,float);  
535 fptr =wrong_arguments_func_pointer_017_func_002;  
  
536 flag = fptr(1,4.5);  
  
537 }  
538  
539 /*
```

Trace

fptr(1,4.5)

02.wo_Defects/wrong_arguments_func_pointer.c:536

```
533 int flag;
534 int (*fptr)(int,float);
535 fptr =wrong_arguments_func_pointer_017_func_002;

536 flag = fptr(1,4.5);

537 }
538
539 /*
```

fptr(1,4.5)

02.wo_Defects/wrong_arguments_func_pointer.c:536

```
533 int flag;
534 int (*fptr)(int,float);
535 fptr =wrong_arguments_func_pointer_017_func_002;

536 flag = fptr(1,4.5);

537 }
538
539 /*
```

02.wo_Defects/zero_division.c:23

Level Low

Status Not processed

```
20 {
21     int dividend = 1000;
22     int ret;
```

```
23     ret = dividend / 1; /*Tool should not detect this line as error*/ /* No ERROR:division
by zero */
```

```
24 }
```

```
25
```

```
26 /*
```

Trace

```
dividend / 1
```

02.wo_Defects/zero_division.c:23

```
20 {
```

```
21   int dividend = 1000;
```

```
22   int ret;
```

```
23   ret = dividend / 1; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */
```

```
24 }
```

```
25
```

```
26 /*
```

```
dividend / 1
```

02.wo_Defects/zero_division.c:23

```
20 {
```

```
21   int dividend = 1000;
```

```
22   int ret;
```

```
23   ret = dividend / 1; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */
```

```
24 }
```

```
25
```

```
26 /*
```

02.wo_Defects/zero_division.c:35

Level Low

Status Not processed

```
32 int dividend = 1000;
33 int ret;
34 dividend /= 1; /*Tool should not detect this line as error*/ /* No ERROR:division by
zero */

35 ret = dividend;

36 }
37
38 /*
```

Trace

```
ret = dividend
```

02.wo_Defects/zero_division.c:35

```
32 int dividend = 1000;
33 int ret;
34 dividend /= 1; /*Tool should not detect this line as error*/ /* No ERROR:
division by zero */

35 ret = dividend;

36 }
37
38 /*
```

```
ret = dividend
```

02.wo_Defects/zero_division.c:35

```
32 int dividend = 1000;
33 int ret;
34 dividend /= 1; /*Tool should not detect this line as error*/ /* No ERROR:
division by zero */

35 ret = dividend;

36 }
37
38 /*
```

02.wo_Defects/zero_division.c:46

Level Low**Status** Not processed

```
43 {  
44     int dividend = 1000;  
45     int ret;  
  
46     ret = dividend % 1; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */  
  
47 }  
48  
49 /*
```

Trace

dividend % 1

02.wo_Defects/zero_division.c:46

```
43 {  
44     int dividend = 1000;  
45     int ret;  
  
46     ret = dividend % 1; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */  
  
47 }  
48  
49 /*
```

dividend % 1

02.wo_Defects/zero_division.c:46

```
43 {  
44     int dividend = 1000;  
45     int ret;  
  
46     ret = dividend % 1; /*Tool should not detect this line as error*/ /* No ERROR:
```

division by zero */

```
47 }  
48  
49 /*
```

02.wo_Defects/zero_division.c:76

Level Low

Status Not processed

```
73 int dividend = 1000;  
74 int divisors[5] = {2, 1, 3, 0, 4};  
75 int ret;
```

76 ret = dividend / divisors[2]; /*Tool should not detect this line as error*/ /* No
ERROR:division by zero */

```
77 }  
78  
79 /*
```

Trace

dividend / divisors[2]

02.wo_Defects/zero_division.c:76

```
73 int dividend = 1000;  
74 int divisors[5] = {2, 1, 3, 0, 4};  
75 int ret;
```

76 ret = dividend / divisors[2]; /*Tool should not detect this line as error*/ /* No
ERROR:division by zero */

```
77 }  
78  
79 /*
```

dividend / divisors[2]

02.wo_Defects/zero_division.c:76

```
73 int dividend = 1000;
74 int divisors[5] = {2, 1, 3, 0, 4};
75 int ret;

76 ret = dividend / divisors[2]; /*Tool should not detect this line as error*/ /* No
ERROR:division by zero */

77 }
78
79 /*
```

02.wo_Defects/zero_division.c:91

Level Low

Status Not processed

```
88 int *p;
89 int ret;
90 p = &zero_division_006_gbl_divisor;

91 ret = dividend / *p; /*Tool should not detect this line as error*/ /* No ERROR:division
by zero */

92 }
93
94 /*
```

Trace

dividend / *p

02.wo_Defects/zero_division.c:91

```
88 int *p;
89 int ret;
90 p = &zero_division_006_gbl_divisor;
```

```
91    ret = dividend / *p; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */
```

```
92 }  
93  
94 /*
```

```
dividend / *p
```

02.wo_Defects/zero_division.c:91

```
88    int *p;  
89    int ret;  
90    p = &zero_division_006_gbl_divisor;
```

```
91    ret = dividend / *p; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */
```

```
92 }  
93  
94 /*
```

02.wo_Defects/zero_division.c:116

Level Low

Status Not processed

```
113 int dividend = 1000;  
114 int ret;  
115 zero_division_007_func_001();
```

```
116 ret = dividend / zero_division_007_s_gbl.divisor; /*Tool should not detect this line  
as error*/ /* No ERROR:division by zero */
```

```
117 }  
118  
119 /*
```

Trace

dividend / zero_division_007_s_gbl.divisor

02.wo_Defects/zero_division.c:116

```
113 int dividend = 1000;
114 int ret;
115 zero_division_007_func_001();

116 ret = dividend / zero_division_007_s_gbl.divisor; /*Tool should not detect this
line as error*/ /* No ERROR:division by zero */

117 }
118
119 /*
```

dividend / zero_division_007_s_gbl.divisor

02.wo_Defects/zero_division.c:116

```
113 int dividend = 1000;
114 int ret;
115 zero_division_007_func_001();

116 ret = dividend / zero_division_007_s_gbl.divisor; /*Tool should not detect this
line as error*/ /* No ERROR:division by zero */

117 }
118
119 /*
```

02.wo_Defects/zero_division.c:127

Level Low**Status** Not processed

```
124 {
125 float dividend = 1000.0;
126 float ret;
```

```
127 ret = dividend / 1.0; /*Tool should not detect this line as error*/ /* No ERROR:
division by zero */
```

```
128 }  
129  
130 /*
```

Trace

```
dividend / 1.0
```

02.wo_Defects/zero_division.c:127

```
124 {  
125 float dividend = 1000.0;  
126 float ret;  
  
127 ret = dividend / 1.0; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

```
128 }  
129  
130 /*
```

```
dividend / 1.0
```

02.wo_Defects/zero_division.c:127

```
124 {  
125 float dividend = 1000.0;  
126 float ret;  
  
127 ret = dividend / 1.0; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

```
128 }  
129  
130 /*
```

02.wo_Defects/zero_division.c:139

Level Low

Status Not processed

```
136 int dividend = 1000;  
137 int divisor = 1;  
138 int ret;
```

```
139 ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */
```

```
140 }  
141  
142 /*
```

Trace

dividend / divisor

02.wo_Defects/zero_division.c:139

```
136 int dividend = 1000;  
137 int divisor = 1;  
138 int ret;
```

```
139 ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

```
140 }  
141  
142 /*
```

dividend / divisor

02.wo_Defects/zero_division.c:139

```
136 int dividend = 1000;  
137 int divisor = 1;  
138 int ret;
```

```
139 ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

```
140 }  
141  
142 /*
```

02.wo_Defects/zero_division.c:154

Level Low

Status Not processed

```
151 divisor = rand();  
152 if (divisor != 0)  
153 {
```

154 ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No
ERROR:division by zero */

```
155 }  
156 }  
157
```

Trace

dividend / divisor

02.wo_Defects/zero_division.c:154

```
151 divisor = rand();  
152 if (divisor != 0)  
153 {
```

154 ret = dividend / divisor; /*Tool should not detect this line as error*/ /*
No ERROR:division by zero */

```
155 }  
156 }  
157
```

dividend / divisor

02.wo_Defects/zero_division.c:154

```
151 divisor = rand();  
152 if (divisor != 0)  
153 {  
  
154         ret = dividend / divisor; /*Tool should not detect this line as error*/ /*  
No ERROR:division by zero */  
  
155     }  
156 }  
157
```

02.wo_Defects/zero_division.c:167

Level Low

Status Not processed

```
164 int dividend = 1000;  
165 int divisor = 2;  
166 int ret;  
  
167 ret = dividend / (2 * divisor - 3); /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */  
  
168 }  
169  
170 /*
```

Trace

dividend / (2 * divisor - 3)

02.wo_Defects/zero_division.c:167

```
164 int dividend = 1000;  
165 int divisor = 2;  
166 int ret;
```

```
167 ret = dividend / (2 * divisor - 3); /*Tool should not detect this line as error*/ /*  
No ERROR:division by zero */
```

```
168 }  
169  
170 /*
```

```
dividend / (2 * divisor - 3)
```

02.wo_Defects/zero_division.c:167

```
164 int dividend = 1000;  
165 int divisor = 2;  
166 int ret;
```

```
167 ret = dividend / (2 * divisor - 3); /*Tool should not detect this line as error*/ /*  
No ERROR:division by zero */
```

```
168 }  
169  
170 /*
```

02.wo_Defects/zero_division.c:179

Level Low

Status Not processed

```
176 int dividend = 1000;  
177 int divisor = 2;  
178 int ret;
```

```
179 ret = dividend / (divisor * divisor - 3); /*Tool should not detect this line as error*/ /*  
No ERROR:division by zero */
```

```
180  
181 }  
182
```

Trace

```
dividend / (divisor * divisor - 3)
```

02.wo_Defects/zero_division.c:179

```
176 int dividend = 1000;  
177 int divisor = 2;  
178 int ret;
```

```
179 ret = dividend / (divisor * divisor - 3); /*Tool should not detect this line as  
error*/ /* No ERROR:division by zero */
```

```
180  
181 }  
182
```

```
dividend / (divisor * divisor - 3)
```

02.wo_Defects/zero_division.c:179

```
176 int dividend = 1000;  
177 int divisor = 2;  
178 int ret;
```

```
179 ret = dividend / (divisor * divisor - 3); /*Tool should not detect this line as  
error*/ /* No ERROR:division by zero */
```

```
180  
181 }  
182
```

02.wo_Defects/zero_division.c:196

Level Low

Status Not processed

```
193 {  
194 int dividend = 1000;  
195 int ret;
```

```
196 ret = dividend / zero_division_013_func_001(); /*Tool should not detect this line as  
error*/ /* No ERROR:division by zero */
```

```
197 }  
198  
199 /*
```

Trace

```
dividend / zero_division_013_func_001()
```

02.wo_Defects/zero_division.c:196

```
193 {  
194 int dividend = 1000;  
195 int ret;  
  
196 ret = dividend / zero_division_013_func_001(); /*Tool should not detect this  
line as error*/ /* No ERROR:division by zero */
```

```
197 }  
198  
199 /*
```

```
dividend / zero_division_013_func_001()
```

02.wo_Defects/zero_division.c:196

```
193 {  
194 int dividend = 1000;  
195 int ret;  
  
196 ret = dividend / zero_division_013_func_001(); /*Tool should not detect this  
line as error*/ /* No ERROR:division by zero */
```

```
197 }  
198  
199 /*
```

```
02.wo_Defects/zero_division.c:207
```

Level Low

Status Not processed

```
204 {  
205 int dividend = 1000;  
206 int ret;
```

```
207 ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No ERROR:  
division by zero */
```

```
208 }  
209  
210 void zero_division_014 ()
```

Trace

dividend / divisor

02.wo_Defects/zero_division.c:207

```
204 {  
205 int dividend = 1000;  
206 int ret;
```

```
207 ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

```
208 }  
209  
210 void zero_division_014 ()
```

dividend / divisor

02.wo_Defects/zero_division.c:207

```
204 {  
205 int dividend = 1000;  
206 int ret;
```

```
207 ret = dividend / divisor; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

```
208 }  
209  
210 void zero_division_014 ()
```

02.wo_Defects/zero_division.c:226

Level Low

Status Not processed

```
223 int divisor1;  
224 int ret;  
225 divisor1 = divisor;
```

226 ret = dividend / divisor1; /*Tool should not detect this line as error*/ /* No ERROR:
division by zero */

```
227 }  
228  
229 /*
```

Trace

dividend / divisor1

02.wo_Defects/zero_division.c:226

```
223 int divisor1;  
224 int ret;  
225 divisor1 = divisor;
```

226 ret = dividend / divisor1; /*Tool should not detect this line as error*/ /* No
ERROR:division by zero */

```
227 }  
228  
229 /*
```

dividend / divisor1

02.wo_Defects/zero_division.c:226

```
223 int divisor1;
224 int ret;
225 divisor1 = divisor;

226 ret = dividend / divisor1; /*Tool should not detect this line as error*/ /* No
ERROR:division by zero */

227 }
228
229 /*
```

02.wo_Defects/zero_division.c:253

Level Low

Status Not processed

```
250 zero_division_016_func_002 ();
251 divisor1 = *zero_division_016_gbl_divisor;
252 divisor2 = divisor1;

253 ret = dividend / divisor2; /*Tool should not detect this line as error*/ /* No ERROR:
division by zero */

254 }
255
256 /*
```

Trace

dividend / divisor2

02.wo_Defects/zero_division.c:253

```
250 zero_division_016_func_002 ();
251 divisor1 = *zero_division_016_gbl_divisor;
252 divisor2 = divisor1;
```

```
253 ret = dividend / divisor2; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

```
254 }  
255  
256 /*
```

dividend / divisor2

02.wo_Defects/zero_division.c:253

```
250 zero_division_016_func_002 ();  
251 divisor1 = *zero_division_016_gbl_divisor;  
252 divisor2 = divisor1;
```

```
253 ret = dividend / divisor2; /*Tool should not detect this line as error*/ /* No  
ERROR:division by zero */
```

```
254 }  
255  
256 /*
```

03.w_Defects_Cpp/improper_error_handling.cpp:22

Level Low

Status Not processed

```
19 try {  
20     int a=0,b=9,c;  
21     if (a==0)  
  
22     c=b/a;  
  
23 }  
24 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error  
handling*/  
25 {
```

Trace

b/a

03.w_Defects_Cpp/improper_error_handling.cpp:22

```
19 try {  
20     int a=0,b=9,c;  
21     if (a==0)  
  
22     c=b/a;  
  
23 }  
24 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error  
handling*/  
25 {
```

b/a

03.w_Defects_Cpp/improper_error_handling.cpp:22

```
19 try {  
20     int a=0,b=9,c;  
21     if (a==0)  
  
22     c=b/a;  
  
23 }  
24 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error  
handling*/  
25 {
```

03.w_Defects_Cpp/improper_error_handling.cpp:42

Level Low**Status** Not processed

```
39 try {  
40     int a=0,b=9,c;  
41     if (a==0) throw(a);  
  
42     c=b/a;
```

```
43 }
44 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error
handling*/
45 {
```

Trace

b/a

03.w_Defects_Cpp/improper_error_handling.cpp:42

```
39 try {
40     int a=0,b=9,c;
41     if (a==0) throw(a);

42     c=b/a;

43 }
44 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error
handling*/
45 {
```

b/a

03.w_Defects_Cpp/improper_error_handling.cpp:42

```
39 try {
40     int a=0,b=9,c;
41     if (a==0) throw(a);

42     c=b/a;

43 }
44 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error
handling*/
45 {
```

03.w_Defects_Cpp/improper_error_handling.cpp:63

Level Low

Status Not processed

```
60 if (a==0.0)
61     throw(a);
62

63 c=b/a;

64 }
65 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error
handling*/
66 {
```

Trace

b/a

03.w_Defects_Cpp/improper_error_handling.cpp:63

```
60 if (a==0.0)
61     throw(a);
62

63 c=b/a;

64 }
65 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error
handling*/
66 {
```

b/a

03.w_Defects_Cpp/improper_error_handling.cpp:63

```
60 if (a==0.0)
61     throw(a);
62

63 c=b/a;

64 }
65 catch(int a) /*Tool should detect this line as error*/ /*ERROR: Improper error
handling*/
```

```
66 {
```

03.w_Defects_Cpp/improper_error_handling.cpp:87

Level Low

Status Not processed

```
84 if (a==0.0)
85     throw(a);
86
```

```
87 c=b/a;
```

```
88 }
89 catch(float a)/*Tool should detect this line as error*/ /*ERROR: Improper error
handling*/
90 {
```

Trace

b/a

03.w_Defects_Cpp/improper_error_handling.cpp:87

```
84 if (a==0.0)
85     throw(a);
86
```

```
87 c=b/a;
```

```
88 }
89 catch(float a)/*Tool should detect this line as error*/ /*ERROR: Improper error
handling*/
90 {
```

b/a

03.w_Defects_Cpp/improper_error_handling.cpp:87

```
84     if (a==0.0)
85         throw(a);
86
87     c=b/a;
88 }
89 catch(float a)/*Tool should detect this line as error*/ /*ERROR: Improper error
handling*/
90 {
```

04.wo_Defects_Cpp/improper_error_handling.cpp:23

Level Low**Status** Not processed

```
20     if (a==0)
21     throw(a);
22
23     c=b/a;
24 }
25 catch(int a) /*Tool should not detect this line as error*/ /*No ERROR: Improper error
handling*/
26 {
```

Trace

b/a

04.wo_Defects_Cpp/improper_error_handling.cpp:23

```
20     if (a==0)
21     throw(a);
22
```

```
23    c=b/a;  
  
24 }  
25 catch(int a) /*Tool should not detect this line as error*/ /*No ERROR: Improper  
error handling*/  
26 {
```

b/a

04.wo_Defects_Cpp/improper_error_handling.cpp:23

```
20    if (a==0)  
21        throw(a);  
22  
  
23    c=b/a;  
  
24 }  
25 catch(int a) /*Tool should not detect this line as error*/ /*No ERROR: Improper  
error handling*/  
26 {
```

04.wo_Defects_Cpp/improper_error_handling.cpp:44

Level Low

Status Not processed

```
41    if (a==0)  
42        throw(a);  
43  
  
44    c=b/a;  
  
45 }  
46 catch(int a) /*Tool should not detect this line as error*/ /*No ERROR: Improper error  
handling*/  
47 {
```

Trace

b/a

04.wo_Defects_Cpp/improper_error_handling.cpp:44

```
41 if (a==0)
42     throw(a);
43
44 c=b/a;
45 }
46 catch(int a) /*Tool should not detect this line as error*/ /*No ERROR: Improper
error handling*/
47 {
```

b/a

04.wo_Defects_Cpp/improper_error_handling.cpp:44

```
41 if (a==0)
42     throw(a);
43
44 c=b/a;
45 }
46 catch(int a) /*Tool should not detect this line as error*/ /*No ERROR: Improper
error handling*/
47 {
```

04.wo_Defects_Cpp/improper_error_handling.cpp:66

Level Low**Status** Not processed

```
63 if (a==0.0)
64     throw(a);
65
```

```
66 c=b/a;
```

```
67 }  
68 catch(float a) /*Tool should not detect this line as error*/ /*No ERROR: Improper error  
handling*/  
69 {
```

Trace

b/a

04.wo_Defects_Cpp/improper_error_handling.cpp:66

```
63 if (a==0.0)  
64     throw(a);  
65  
  
66 c=b/a;
```

```
67 }  
68 catch(float a) /*Tool should not detect this line as error*/ /*No ERROR:  
Improper error handling*/  
69 {
```

b/a

04.wo_Defects_Cpp/improper_error_handling.cpp:66

```
63 if (a==0.0)  
64     throw(a);  
65  
  
66 c=b/a;
```

```
67 }  
68 catch(float a) /*Tool should not detect this line as error*/ /*No ERROR:  
Improper error handling*/  
69 {
```

04.wo_Defects_Cpp/improper_error_handling.cpp:91

Level Low

Status Not processed

```
88 if (a==0.0)
89     throw(a);
90

91 c=b/a;

92 }
93 catch(float a) /*Tool should not detect this line as error*/ /*No ERROR: Improper error
handling*/
94 {
```

Trace

b/a

04.wo_Defects_Cpp/improper_error_handling.cpp:91

```
88 if (a==0.0)
89     throw(a);
90

91 c=b/a;

92 }
93 catch(float a) /*Tool should not detect this line as error*/ /*No ERROR:
Improper error handling*/
94 {
```

b/a

04.wo_Defects_Cpp/improper_error_handling.cpp:91

```
88 if (a==0.0)
89     throw(a);
90

91 c=b/a;

92 }
93 catch(float a) /*Tool should not detect this line as error*/ /*No ERROR:
Improper error handling*/
```

```
94 {
```

Garbage memory usage is possible (C/C++)

Description

The application probably uses garbage memory. This may lead to undefined behavior of the application.

Example

In the following example, a pointer returned from a function that points to a local variable in the function stack is used. The data for this pointer outside the function is garbage.

```
int* func()
{
    int val;
    int *ptr = &val;
    return ptr;
}

int main()
{
    int *p = fun();
    printf("%d", *p); // Garbage pointer
    ...
    return 0;
}
```

In the following example, the pointer obtained from malloc(0) is used. The behavior is implementation-defined: the value returned shall be either a null pointer or a unique pointer.

```
void func2()
{
    int *p = malloc(0);
```

```
*p = 0;  
}
```

Recommendations

- Do not use a pointer that points to a local variable on the function stack outside that function.
- Do not use malloc(0) if possible.

Links

1. Dangling, Void , Null and Wild Pointers
2. Top 20 C pointer mistakes and how to fix them
3. malloc

Vulnerability Entries

01.w_Defects/littlemem_st.c:93

Level Low

Status Not processed

```
90  
91 littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;  
92 littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should detect this line as error*/  
/*ERROR:Little Memory or Overflow*/
```

93 }

```
94  
95 /*  
96 * Types of defects: Allocate small size for type - structure (static)
```

Trace

vflag == 1

01.w_Defects/littlemem_st.c:360

```
357 extern volatile int vflag;
358 void littlemem_st_main ()
359 {
360     if (vflag == 1 || vflag == 888)
361     {
362         littlemem_st_001();
363     }
```

char buf[10]

01.w_Defects/littlemem_st.c:93

```
90
91     littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;
92     littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should detect this line as
error*/ /*ERROR:Little Memory or Overflow*/
93 }

94
95 /*
96 * Types of defects: Allocate small size for type - structure (static)
```

01.w_Defects/return_local.c:19

Level Low

Status Not processed

```
16 int* return_local_001_func_001 ()
17 {
18     int buf[5];
```

```
19     return buf; /*Tool should detect this line as error*/ /*ERROR: return - pointer to local
variable */
```

```
20 }
21
22 void return_local_001 ()
```

Trace

vflag == 1

01.w_Defects/return_local.c:53

```
50 extern volatile int vflag;
51 void return_local_main ()
52 {

53     if (vflag == 1 || vflag ==888)

54     {
55         return_local_001();
56     }
```

int buf[5]

01.w_Defects/return_local.c:19

```
16 int* return_local_001_func_001 ()
17 {
18     int buf[5];

19     return buf; /*Tool should detect this line as error*/ /*ERROR: return - pointer
to local variable */

20 }
21
22 void return_local_001 ()
```

01.w_Defects/st_cross_thread_access.c:55

Level Low

Status Not processed

```
52     unsigned long ip = (unsigned long)pthread_self();  
53     printf("Task1! Cross thread stack access, threadID# %lu! thread no =%s %d\n",ip ,  
th,*st_cross_thread_access_001_glb_ptr);  
54 #endif /* defined(CHECKER_POLYSPACE) */  
  
55     return NULL;  
  
56 }  
57  
58 void * st_cross_thread_access_001_tsk_002 (void *pram)
```

Trace

```
int arr[10]
```

01.w_Defects/st_cross_thread_access.c:55

```
52     unsigned long ip = (unsigned long)pthread_self();  
53     printf("Task1! Cross thread stack access, threadID# %lu! thread no =%s %  
d\n",ip ,th,*st_cross_thread_access_001_glb_ptr);  
54 #endif /* defined(CHECKER_POLYSPACE) */  
  
55     return NULL;  
  
56 }  
57  
58 void * st_cross_thread_access_001_tsk_002 (void *pram)
```

```
int arr[10]
```

01.w_Defects/st_cross_thread_access.c:55

```
52     unsigned long ip = (unsigned long)pthread_self();  
53     printf("Task1! Cross thread stack access, threadID# %lu! thread no =%s %  
d\n",ip ,th,*st_cross_thread_access_001_glb_ptr);  
54 #endif /* defined(CHECKER_POLYSPACE) */  
  
55     return NULL;  
  
56 }  
57  
58 void * st_cross_thread_access_001_tsk_002 (void *pram)
```

01.w_Defects/st_cross_thread_access.c:143

Level Low**Status** Not processed

```
140     pthread_mutex_unlock(&st_cross_thread_access_002_glb_mutex);
141 }
142 #endif /* defined(CHECKER_POLYSPACE) */

143 return NULL;

144 }
145
146 void * st_cross_thread_access_002_tsk_002 (void * pram)
```

Trace

```
st_cross_thread_access_002_var ==
(intptr_t)pram
```

01.w_Defects/st_cross_thread_access.c:124

```
121 {
122 #if !defined(CHECKER_POLYSPACE)
123 int arr[5] = {10,20,30,40,50};

124 if(st_cross_thread_access_002_var == (intptr_t)pram)

125 {
126     pthread_mutex_lock(&st_cross_thread_access_002_glb_mutex);
127     st_cross_thread_access_002_glb_data =
(st_cross_thread_access_002_glb_data % 100) + 1;
```

```
int arr[5] = {10,20,30,40,50}
```

01.w_Defects/st_cross_thread_access.c:143

```
140     pthread_mutex_unlock(&st_cross_thread_access_002_glb_mutex);
141 }
142 #endif /* defined(CHECKER_POLYSPACE) */

143 return NULL;
```

```
144 }
145
146 void * st_cross_thread_access_002_tsk_002 (void * pram)
```

01.w_Defects/st_cross_thread_access.c:231

Level Low**Status** Not processed

```
228     pthread_mutex_unlock (&st_cross_thread_access_003_glb_mutex);
229 }
230 #endif /* ! defined(CHECKER_POLYSPACE) */

231 return NULL;

232 }
233
234 int st_cross_thread_access_003_func_002 (int a )
```

Trace

float fptr = 50.2

01.w_Defects/st_cross_thread_access.c:231

```
228     pthread_mutex_unlock (&st_cross_thread_access_003_glb_mutex);
229 }
230 #endif /* ! defined(CHECKER_POLYSPACE) */

231 return NULL;

232 }
233
234 int st_cross_thread_access_003_func_002 (int a )
```

```
float fptr = 50.2
```

01.w_Defects/st_cross_thread_access.c:231

```
228     pthread_mutex_unlock (&st_cross_thread_access_003_glb_mutex);
229 }
230 #endif /* ! defined(CHECKER_POLYSPACE) */

231 return NULL;

232 }
233
234 int st_cross_thread_access_003_func_002 (int a )
```

01.w_Defects/st_cross_thread_access.c:320

Level Low

Status Not processed

```
317
318 #endif /* defined(CHECKER_POLYSPACE) */
319

320 return NULL;

321 }
322
323 void * st_cross_thread_access_004_tsk_002(void *input)
```

Trace

```
ip >= 0
```

01.w_Defects/st_cross_thread_access.c:306

```
303
304 while (i>0)
305 {
```

```
306     if (ip >= 0)
```

```
307  {
308      pthread_mutex_lock( &st_cross_thread_access_004_glb_mutex_1
309 );
310      pbuf[0] = &buf11;
```

```
char *buf11="String111"
```

01.w_Defects/st_cross_thread_access.c:320

```
317
318 #endif /* defined(CHECKER_POLYSPACE) */
319

320 return NULL;

321 }
322
323 void * st_cross_thread_access_004_tsk_002(void *input)
```

01.w_Defects/st_cross_thread_access.c:399

Level Low

Status Not processed

```
396
397 #endif /* defined(CHECKER_POLYSPACE) */
398 }

399 return NULL;

400 }
401
402 void * st_cross_thread_access_005_tsk_002 (void *pram)
```

Trace

```
st_cross_thread_access_005_thread_set !=  
NO_THREAD
```

01.w_Defects/st_cross_thread_access.c:379

```
376 void * st_cross_thread_access_005_tsk_001 (void *pram)  
377 {  
378 #if !defined(CHECKER_POLYSPACE)  
  
379 if (st_cross_thread_access_005_thread_set != NO_THREAD)  
  
380 {  
381 ;  
382 }
```

```
int arr1[4][4] = {{1,2,3,4}, {11,22,33,44},  
{33,44,55,66}, {55,66,77,88}}
```

01.w_Defects/st_cross_thread_access.c:399

```
396  
397 #endif /* defined(CHECKER_POLYSPACE) */  
398 }  
  
399 return NULL;  
  
400 }  
401  
402 void * st_cross_thread_access_005_tsk_002 (void *pram)
```

01.w_Defects/st_cross_thread_access.c:482

Level Low

Status Not processed

```
479 printf("Task6_1! Cross thread stack access, thread #%Id! aa=%d\n",ip,sprt->a);  
480 #endif /* defined(PRINT_DEBUG) */  
481 #endif /* defined(CHECKER_POLYSPACE) */
```

```
482 return NULL;
```

```
483 }  
484  
485 void * st_cross_thread_access_006_tsk_002(void *input)
```

Trace

```
ip >= 0
```

01.w_Defects/st_cross_thread_access.c:470

```
467 st_cross_thread_access_006_s_001 s ={1,2};  
468 for (i=0;i<=4;i++)  
469 {  
  
470 if (ip >= 0)  
  
471 {  
472     pthread_mutex_lock( &st_cross_thread_access_006_glb_mutex_1 );  
473     sprt = &s;
```

```
st_cross_thread_access_006_s_001 s =  
{1,2}
```

01.w_Defects/st_cross_thread_access.c:482

```
479 printf("Task6_1! Cross thread stack access, thread #%ld! aa=%d\n",ip,sprt->a);  
480 #endif /* defined(PRINT_DEBUG) */  
481 #endif /* defined(CHECKER_POLYSPACE) */
```

```
482 return NULL;
```

```
483 }  
484  
485 void * st_cross_thread_access_006_tsk_002(void *input)
```

02.wo_Defects/littlemem_st.c:94

Level Low

Status Not processed

```
91
92     littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;
93     littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should not detect this line as error*/
/*No ERROR:Little Memory or Overflow*/

94 }

95
96 /*
97 * Types of defects: Allocate small size for type - structure (static)
```

Trace

```
vflag == 1
```

```
02.wo_Defects/littlemem_st.c:361
```

```
358 extern volatile int vflag;
359 void littlemem_st_main ()
360 {
361     if (vflag == 1 || vflag ==888)
362     {
363         littlemem_st_001();
364     }
```

```
char buf[12]
```

```
02.wo_Defects/littlemem_st.c:94
```

```
91
92     littlemem_st_004_s_001_gbl_str = (littlemem_st_004_s_001 *)buf;
93     littlemem_st_004_s_001_gbl_str->c = 1; /*Tool should not detect this line as
error*/ /*No ERROR:Little Memory or Overflow*/

94 }

95
96 /*
97 * Types of defects: Allocate small size for type - structure (static)
```

02.wo_Defects/st_cross_thread_access.c:56

Level Low**Status** Not processed

```
53 unsigned long ip = (unsigned long)pthread_self();
54 printf("Task1! Cross thread stack access, threadID# %lu! thread no = %s %d\n",ip ,
th,*st_cross_thread_access_001_glb_ptr);
55 #endif /* defined(CHECKER_POLYSPACE) */

56 return NULL;

57 }
58
59 void * st_cross_thread_access_001_tsk_002 (void *pram)
```

Trace

int arr[10]

02.wo_Defects/st_cross_thread_access.c:56

```
53 unsigned long ip = (unsigned long)pthread_self();
54 printf("Task1! Cross thread stack access, threadID# %lu! thread no = %s %
d\n",ip ,th,*st_cross_thread_access_001_glb_ptr);
55 #endif /* defined(CHECKER_POLYSPACE) */

56 return NULL;

57 }
58
59 void * st_cross_thread_access_001_tsk_002 (void *pram)
```

int arr[10]

02.wo_Defects/st_cross_thread_access.c:56

```
53 unsigned long ip = (unsigned long)pthread_self();
54 printf("Task1! Cross thread stack access, threadID# %lu! thread no = %s %
d\n",ip ,th,*st_cross_thread_access_001_glb_ptr);
55 #endif /* defined(CHECKER_POLYSPACE) */

56 return NULL;
```

```
57 }  
58  
59 void * st_cross_thread_access_001_tsk_002 (void *pram)
```

02.wo_Defects/st_cross_thread_access.c:144

Level Low

Status Not processed

```
141     pthread_mutex_unlock(&st_cross_thread_access_002_glb_mutex);  
142 }  
143 #endif /* defined(CHECKER_POLYSPACE) */  
  
144 return NULL;  
  
145 }  
146  
147 void * st_cross_thread_access_002_tsk_002 (void * pram)
```

Trace

```
st_cross_thread_access_002_var ==  
(intptr_t)pram
```

02.wo_Defects/st_cross_thread_access.c:125

```
122 {  
123 #if !defined(CHECKER_POLYSPACE)  
124 int arr[5] = {10,20,30,40,50};  
  
125 if(st_cross_thread_access_002_var == (intptr_t)pram)  
  
126 {  
127     pthread_mutex_lock(&st_cross_thread_access_002_glb_mutex);  
128     st_cross_thread_access_002_glb_data =  
     (st_cross_thread_access_002_glb_data % 100) + 1;
```

```
int arr[5] = {10,20,30,40,50}
```

02.wo_Defects/st_cross_thread_access.c:144

```
141     pthread_mutex_unlock(&st_cross_thread_access_002_glb_mutex);
142 }
143 #endif /* defined(CHECKER_POLYSPACE) */

144 return NULL;

145 }
146
147 void * st_cross_thread_access_002_tsk_002 (void * pram)
```

02.wo_Defects/st_cross_thread_access.c:233

Level Low

Status Not processed

```
230     pthread_mutex_unlock (&st_cross_thread_access_003_glb_mutex);
231 }
232 #endif /* ! defined(CHECKER_POLYSPACE) */

233 return NULL;

234 }
235
236 int st_cross_thread_access_003_func_002 (int a )
```

Trace

```
float fptr = 50.2
```

02.wo_Defects/st_cross_thread_access.c:233

```
230     pthread_mutex_unlock (&st_cross_thread_access_003_glb_mutex);
231 }
232 #endif /* ! defined(CHECKER_POLYSPACE) */

233 return NULL;
```

```
234 }  
235  
236 int st_cross_thread_access_003_func_002 (int a )
```

```
float fptr = 50.2
```

02.wo_Defects/st_cross_thread_access.c:233

```
230     pthread_mutex_unlock (&st_cross_thread_access_003_glb_mutex);  
231 }  
232 #endif /* ! defined(CHECKER_POLYSPACE) */
```

```
233 return NULL;
```

```
234 }  
235  
236 int st_cross_thread_access_003_func_002 (int a )
```

02.wo_Defects/st_cross_thread_access.c:321

Level Low

Status Not processed

```
318  
319 #endif /* defined(CHECKER_POLYSPACE) */  
320
```

```
321 return NULL;
```

```
322 }  
323  
324 void * st_cross_thread_access_004_tsk_002(void *input)
```

Trace

```
ip >= 0
```

02.wo_Defects/st_cross_thread_access.c:307

```
304
305 while (i>0)
306 {

307     if (ip >= 0)

308     {
309         pthread_mutex_lock( &st_cross_thread_access_004_glb_mutex_1
);
310         pbuf[0] = &buf11;
```

```
char *buf11="String111"
```

02.wo_Defects/st_cross_thread_access.c:321

```
318
319 #endif /* defined(CHECKER_POLYSPACE) */
320

321     return NULL;

322 }
323
324 void * st_cross_thread_access_004_tsk_002(void *input)
```

02.wo_Defects/st_cross_thread_access.c:400

Level Low

Status Not processed

```
397
398 #endif /* defined(CHECKER_POLYSPACE) */
399 }
```

```
400     return NULL;
```

```
401 }
402
403 void * st_cross_thread_access_005_tsk_002 (void *pram)
```

Trace

```
└─ st_cross_thread_access_005_thread_set !=  
    NO_THREAD
```

```
02.wo_Defects/st_cross_thread_access.c:380
```

```
377 void * st_cross_thread_access_005_tsk_001 (void *pram)
378 {
379 #if !defined(CHECKER_POLYSPACE)
```

```
380     if (st_cross_thread_access_005_thread_set != NO_THREAD)
```

```
381     {
382         ;
383     }
```

```
int arr1[4][4] = {{1,2,3,4}, {11,22,33,44},
{33,44,55,66}, {55,66,77,88}}
```

```
02.wo_Defects/st_cross_thread_access.c:400
```

```
397
398 #endif /* defined(CHECKER_POLYSPACE) */
399 }
```

```
400     return NULL;
```

```
401 }
402
403 void * st_cross_thread_access_005_tsk_002 (void *pram)
```

```
02.wo_Defects/st_cross_thread_access.c:483
```

Level Low

Status Not processed

```
480 printf("Task6_1! Cross thread stack access, thread #%ld! aa=%d\n",ip,sprt->a);
481 #endif /* defined(PRINT_DEBUG) */
482 #endif /* defined(CHECKER_POLYSPACE) */

483 return NULL;

484 }
485
486 void * st_cross_thread_access_006_tsk_002(void *input)
```

Trace

```
ip >= 0
```

```
02.wo_Defects/st_cross_thread_access.c:471
```

```
468 st_cross_thread_access_006_s_001 s ={1,2};
469 for (i=0;i<=4;i++)
470 {

471 if (ip >= 0)

472 {
473     pthread_mutex_lock( &st_cross_thread_access_006_glb_mutex_1 );
474     sprt = &s;
```

```
st_cross_thread_access_006_s_001 s =
{1,2}
```

```
02.wo_Defects/st_cross_thread_access.c:483
```

```
480 printf("Task6_1! Cross thread stack access, thread #%ld! aa=%d\n",ip,sprt-
>a);
481 #endif /* defined(PRINT_DEBUG) */
482 #endif /* defined(CHECKER_POLYSPACE) */

483 return NULL;

484 }
485
486 void * st_cross_thread_access_006_tsk_002(void *input)
```

Incorrect function call (C/C++)

Description

Function call with incorrect parameters.

Probably, there are incorrect number of argument, incorrect order of arguments or incorrect argument type, which may lead to undefined behavior of application.

Example

In the following example, the delete operator is called with an uninitialized argument:

```
int *x;  
delete x;
```

The correct version:

```
int *x = new int;  
delete x;
```

Recommendations

- Make sure the function is called correctly.

Links

1. CWE-686: Function Call With Incorrect Argument Type
2. CWE-683: Function Call With Incorrect Order of Arguments
3. CWE-688: Function Call With Incorrect Variable or Reference as Argument
4. CWE-685: Function Call With Incorrect Number of Arguments
5. CWE-687: Function Call With Incorrectly Specified Argument Value

Vulnerability Entries

01.w_Defects/func_pointer.c:375

Level Low

Status Not processed

```
372 int ret;
373 func = (int (*)(void))func_pointer_004_func_001;
374 func1 = func;
```

375 ret = func1();/*Tool should detect this line as error*/ /*ERROR:Bad function pointer casting*/

```
376 }
377
378 /*
```

Trace

vflag == 1

01.w_Defects/func_pointer.c:617

```
614 extern volatile int vflag;
615 void func_pointer_main ()
616 {
```

617 if (vflag == 1 || vflag ==888)

```
618 {
619     func_pointer_001();
620 }
```

func1()

01.w_Defects/func_pointer.c:375

```
372 int ret;
373 func = (int (*)(void))func_pointer_004_func_001;
374 func1 = func;
```

```
375 ret = func1();/*Tool should detect this line as error*/ /*ERROR:Bad function  
pointer casting*/  
  
376 }  
377  
378 /*
```

01.w_Defects/memory_allocation_failure.c:514

Level Low

Status Not processed

```
511     free(dptr);  
512     dptr = NULL;  
513 }  
  
514 printf("%d",a);  
  
515 sink = b;  
516 }  
517
```

Trace

vflag == 1

01.w_Defects/memory_allocation_failure.c:724

```
721 extern volatile int vflag;  
722 void memory_allocation_failure_main ()  
723 {  
  
724 if (vflag == 1 || vflag ==888)  
  
725 {  
726     memory_allocation_failure_001();  
727 }
```

```
printf("%d",a)
```

01.w_Defects/memory_allocation_failure.c:514

```
511     free(dptr);
512     dptr = NULL;
513 }

514 printf("%d",a);

515 sink = b;
516 }
517
```

01.w_Defects/uninit_memory_access.c:27

Level Low

Status Not processed

```
24 unsigned long a;
25 unsigned long *ret;
26 ret = &a;
```

```
27 printf("%ld ",*ret);/*Tool should detect this line as error*/ /*ERROR:Uninitialized
Memory Access*/
```

```
28 }
29
30 /*
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_memory_access.c:462

```
459 extern volatile int vflag;
460 void uninit_memory_access_main ()
461 {
```

```
462 if (vflag == 1 || vflag ==888)

463 {
464     uninit_memory_access_001();
465 }
```

*ret

01.w_Defects/uninit_memory_access.c:27

```
24 unsigned long a;
25 unsigned long *ret;
26 ret = &a;

27 printf("%ld ",*ret);/*Tool should detect this line as error*/ /*ERROR:
   Uninitialized Memory Access*/

28 }
29
30 /*
```

01.w_Defects/uninit_memory_access.c:54

Level Low

Status Not processed

```
51 char *str2 ;
52 if (str1!=NULL)
53 {

54     strcpy(str1, str2);

55     printf("%s %s\n",str1,str2);/*Tool should detect this line as error*/ /*ERROR:
   Uninitialized Memory Access*/
56     free(str1);
57 }
```

Trace

vflag == 1

01.w_Defects/uninit_memory_access.c:462

```
459 extern volatile int vflag;
460 void uninit_memory_access_main ()
461 {
462     if (vflag == 1 || vflag == 888)
463     {
464         uninit_memory_access_001();
465     }
```

strcpy(str1, str2)

01.w_Defects/uninit_memory_access.c:54

```
51 char *str2 ;
52 if (str1!=NULL)
53 {
54     strcpy(str1, str2);
55     printf("%s %s\n",str1,str2);/*Tool should detect this line as error*/
/*ERROR:Uninitialized Memory Access*/
56     free(str1);
57 }
```

01.w_Defects/uninit_memory_access.c:127

Level Low

Status Not processed

```
124 {
125     char *str1 = (char *) calloc(25,sizeof(char));
126     char *str2 ;
127     uninit_memory_access_006_func_001(str1, str2);
```

```
128 printf("%s\n", str1);/*Tool should detect this line as error*/ /*ERROR:Uninitialized  
Memory Access*/  
129 }  
130
```

Trace

```
vflag == 1
```

```
01.w_Defects/uninit_memory_access.c:462
```

```
459 extern volatile int vflag;  
460 void uninit_memory_access_main ()  
461 {  
  
462 if (vflag == 1 || vflag ==888)  
  
463 {  
464     uninit_memory_access_001();  
465 }
```

```
uninit_memory_access_006_func_001(str1,  
str2)
```

```
01.w_Defects/uninit_memory_access.c:127
```

```
124 {  
125     char *str1 = (char *) calloc(25,sizeof(char));  
126     char *str2 ;  
  
127     uninit_memory_access_006_func_001(str1, str2);  
  
128     printf("%s\n", str1);/*Tool should detect this line as error*/ /*ERROR:  
Uninitialized Memory Access*/  
129 }  
130
```

```
01.w_Defects/uninit_memory_access.c:319
```

Level Low

Status Not processed

```
316 void uninit_memory_access_011 ()  
317 {  
318     char *str ;  
  
319     uninit_memory_access_011_func_001(str);/*Tool should detect this line as error*/  
/*ERROR:Uninitialized Memory Access*/  
  
320 }  
321  
322 /*
```

Trace

```
vflag == 1
```

```
01.w_Defects/uninit_memory_access.c:462
```

```
459 extern volatile int vflag;  
460 void uninit_memory_access_main ()  
461 {  
  
462     if (vflag == 1 || vflag == 888)  
  
463     {  
464         uninit_memory_access_001();  
465     }
```

```
uninit_memory_access_011_func_001(str);  
/*Tool should detect this line as error*/  
/*ERROR:Uninitialized Memory Access*/
```

```
01.w_Defects/uninit_memory_access.c:319
```

```
316 void uninit_memory_access_011 ()  
317 {  
318     char *str ;  
  
319     uninit_memory_access_011_func_001(str);/*Tool should detect this line as  
error*/ /*ERROR:Uninitialized Memory Access*/  
  
320 }
```

```
321
322 /*
```

01.w_Defects/uninit_pointer.c:71

Level Low

Status Not processed

```
68 {
69   int a = 0;
70   int *p ;
```

71 uninit_pointer_004_func_001(p);/*Tool should detect this line as error*/ /*ERROR:
Uninitialized pointer*/

```
72 }
73
74 /*
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {
```

421 if (vflag == 1 || vflag ==888)

```
422 {
423     uninit_pointer_001();
424 }
```

```
uninit_pointer_004_func_001(p);/*Tool  
should detect this line as error*/ /*ERROR:  
Uninitialized pointer*/
```

01.w_Defects/uninit_pointer.c:71

```
68 {  
69   int a = 0;  
70   int *p ;  
  
71   uninit_pointer_004_func_001(p);/*Tool should detect this line as error*/  
/*ERROR:Uninitialized pointer*/  
  
72 }  
73  
74 /*
```

01.w_Defects/uninit_pointer.c:187

Level Low

Status Not processed

```
184 }  
185 if(uninit_pointer_009_func_001(flag)>0)  
186 {  
  
187     strcpy(buf1,buf);/*Tool should detect this line as error*/ /*ERROR:  
Uninitialized pointer*/  
  
188 }  
189 }  
190
```

Trace

vflag == 1

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;
419 void uninit_pointer_main ()
420 {
421     if (vflag == 1 || vflag == 888)
422     {
423         uninit_pointer_001();
424     }
```

strcpy(buf1,buf)

01.w_Defects/uninit_pointer.c:187

```
184 }
185     if(uninit_pointer_009_func_001(flag)>0)
186     {
187         strcpy(buf1,buf); /*Tool should detect this line as error*/ /*ERROR:
188         Uninitialized pointer*/
189     }
190
```

01.w_Defects/uninit_pointer.c:358

Level Low

Status Not processed

```
355 void uninit_pointer_015 ()
```

```
356 {
```

```
357     int *ptr;
```

```
358     uninit_pointer_015_func_001(ptr); /*Tool should detect this line as error*/
/*ERROR:Uninitialized pointer*/
```

```
359 }  
360  
361 /*
```

Trace

```
vflag == 1
```

01.w_Defects/uninit_pointer.c:421

```
418 extern volatile int vflag;  
419 void uninit_pointer_main ()  
420 {  
  
421 if (vflag == 1 || vflag ==888)  
  
422 {  
423     uninit_pointer_001();  
424 }
```

uninit_pointer_015_func_001(ptr);/*Tool
should detect this line as error*/ /*ERROR:
Uninitialized pointer*/

01.w_Defects/uninit_pointer.c:358

```
355 void uninit_pointer_015 ()  
356 {  
357     int *ptr;  
  
358     uninit_pointer_015_func_001(ptr);/*Tool should detect this line as error*/  
/*ERROR:Uninitialized pointer*/  
  
359 }  
360  
361 /*
```

01.w_Defects/wrong_arguments_func_pointer.c:161

Level Low

Status Not processed

```
158 int (*func)(int);
159 int ret;
160 func = (int (*)(int))wrong_arguments_func_pointer_006_func_001;

161 ret = func(5);/*Tool should detect this line as error//ERROR:Wrong arguments
passed to a function pointer*/

162
163 }
164
```

Trace

```
vflag == 1
```

```
01.w_Defects/wrong_arguments_func_pointer.c:610
```

```
607 extern volatile int vflag;
608 void wrong_arguments_func_pointer_main ()
609 {

610 if (vflag == 1 || vflag == 888)

611 {
612     wrong_arguments_func_pointer_001();
613 }
```

```
func(5)
```

```
01.w_Defects/wrong_arguments_func_pointer.c:161
```

```
158 int (*func)(int);
159 int ret;
160 func = (int (*)(int))wrong_arguments_func_pointer_006_func_001;

161 ret = func(5);/*Tool should detect this line as error//ERROR:Wrong
arguments passed to a function pointer*/

162
163 }
164
```

01.w_Defects/wrong_arguments_func_pointer.c:381

Level Low**Status** Not processed

```
378 char *str1=NULL;
379 void (*fptr)(char *);
380 fptr = (void (*)(char*))wrong_arguments_func_pointer_013_func_001;
```

381 fptr(str1);/*Tool should detect this line as error**/*ERROR:Wrong arguments passed to a function pointer*/

```
382 strcpy(str1,str);
383 free(str1);
384 str1 = NULL;
```

Trace

vflag == 1

01.w_Defects/wrong_arguments_func_pointer.c:610

```
607 extern volatile int vflag;
608 void wrong_arguments_func_pointer_main ()
609 {
610     if (vflag == 1 || vflag ==888)
611     {
612         wrong_arguments_func_pointer_001();
613     }
}
```

```
fptr(str1)
```

01.w_Defects/wrong_arguments_func_pointer.c:381

```
378 char *str1=NULL;  
379 void (*fptr)(char *);  
380 fptr = (void (*)(char*))wrong_arguments_func_pointer_013_func_001;
```

381 fptr(str1);/*Tool should detect this line as error**/*ERROR:Wrong arguments passed to a function pointer*/

```
382 strcpy(str1,str);  
383 free(str1);  
384 str1 = NULL;
```

01.w_Defects/wrong_arguments_func_pointer.c:423

Level Low

Status Not processed

```
420     long (*fptr)(float *);  
421     long a;  
422     fptr = (long (*)(float * ))wrong_arguments_func_pointer_014_func_002;
```

423 a =fptr(&f);/*Tool should detect this line as error**/*ERROR:Wrong arguments passed to a function pointer*/

```
424 }  
425  
426 }
```

Trace

```
vflag == 1
```

01.w_Defects/wrong_arguments_func_pointer.c:610

```
607 extern volatile int vflag;  
608 void wrong_arguments_func_pointer_main ()  
609 {
```

```
610 if (vflag == 1 || vflag ==888)

611 {
612     wrong_arguments_func_pointer_001();
613 }
```

fptr(&f)

01.w_Defects/wrong_arguments_func_pointer.c:423

```
420     long (*fptr)(float *);
421     long a;
422     fptr = (long (*)(float *)) 
wrong_arguments_func_pointer_014_func_002;

423         a =fptr(&f);/*Tool should detect this line as error//ERROR:Wrong
arguments passed to a function pointer*/

424 }
425
426 }
```

01.w_Defects/wrong_arguments_func_pointer.c:457

Level Low

Status Not processed

```
454 {
455 void (*fptr)(char **);
456 fptr = (void (*)(char**))wrong_arguments_func_pointer_015_func_002;

457 fptr(wrong_arguments_func_pointer_015_dst1_gbl);/*Tool should detect this line
as error//ERROR:Wrong arguments passed to a function pointer*/

458 break;
459 }
460 for(i=0;i<5;i++)
```

Trace

```
vflag == 1
```

01.w_Defects/wrong_arguments_func_pointer.c:610

```
607 extern volatile int vflag;
608 void wrong_arguments_func_pointer_main ()
609 {
610     if (vflag == 1 || vflag == 888)
611     {
612         wrong_arguments_func_pointer_001();
613     }
}
```

```
fptr
(wrong_arguments_func_pointer_015_dst1_
gbl)
```

01.w_Defects/wrong_arguments_func_pointer.c:457

```
454 {
455     void (*fptr)(char **);
456     fptr = (void (*)(char **))wrong_arguments_func_pointer_015_func_002;

457     fptr(wrong_arguments_func_pointer_015_dst1_gbl); /*Tool should detect
this line as error*/ /*ERROR:Wrong arguments passed to a function pointer*/

458     break;
459 }
460 for(i=0;i<5;i++)
```

Use of overlapping buffers (C/C++)

Description

Overlapping buffers are passed to the function that works with strings or memory. Undefined behavior of application is possible.

Example

In the following example, the `memcpy` function on overlapping buffers is used:

```
extern void* a;
memcpy(a+2, a+1, 8);
```

Specification says that behavior of memcpy on overlapping buffers is not defined.

Recommendations

- Do not use overlapping buffers when working with strings or memory.

Links

1. memcpy of overlapping buffers

Vulnerability Entries

01.w_Defects/wrong_arguments_func_pointer.c:213

Level Low

Status Not processed

```
210 */
211 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2, char*str3)
212 {

213     strcat(str1,str2);

214     strcpy(str3,str1);
215     return ('c');
216 }
```

Trace

vflag == 1

01.w_Defects/wrong_arguments_func_pointer.c:610

```
607 extern volatile int vflag;
608 void wrong_arguments_func_pointer_main ()
609 {
```

```
610     if (vflag == 1 || vflag ==888)
```

```
611 {  
612     wrong_arguments_func_pointer_001();  
613 }
```

```
strcat(str1,str2) strcat(str1,str2)
```

01.w_Defects/wrong_arguments_func_pointer.c:213

```
210 */  
211 char wrong_arguments_func_pointer_009_func_001(char *str1, char *str2,  
char*str3)  
212 {  
  
213     strcat(str1,str2);  
  
214     strcpy(str3,str1);  
215     return ('c');  
216 }
```

Using an insecure method (C/C++)

Description

There is a safer analogue to the function to be called. For example, for many functions whose execution may lead to buffer overflow, there are analogues that check the buffer size: use PathCchAppend instead of PathAppend, memcpys instead of memcpy.

Example

In the following example, the application uses PathAppend function:

```
TCHAR pszPath[MAX_PATH];  
TCHAR Pf[MAX_PATH];  
PathAppend(pszPath, Pf);
```

Recommendations

- Use PathCchAppend instead of PathAppend, memcpys instead of memcpy.

Links

1. CWE-676: Use of Potentially Dangerous Function

Vulnerability Entries

01.w_Defects/deletion_of_data_structure_sentinel.c:41

Level Low

Status Not processed

```
38 {  
39     char str1[]{"This is a string";  
40     char str2[16];
```

```
41     memcpy(str2,str1,strlen(str1)); /*Tool should detect this line as error*/ /*ERROR:  
Deletion of a data structure sentinel*/
```

```
42 }  
43  
44 /*
```

01.w_Defects/free_null_pointer.c:108

Level Low

Status Not processed

```
105 void free_null_pointer_005 ()  
106 {  
107     char *str = "This is a string";
```

```
108     free_null_pointer_005_func_001(strlen(str));
```

```
109     strcpy(free_null_pointer_005_gbl_ptr,str);  
110     free(free_null_pointer_005_gbl_ptr); /* Tool should detect this line as error  
/*/*ERROR:Freeing a NULL pointer*/  
111     free_null_pointer_005_gbl_ptr = NULL;
```

01.w_Defects/free_null_pointer.c:240

Level Low

Status Not processed

```
237 {  
238     char *str = "This is a string";  
239     char *str1=NULL;  
  
240     free_null_pointer_008_func_001(strlen(str),&str1);  
  
241     strcpy(str1,str);  
242     free(str1);/* Tool should detect this line as error *//*ERROR:Freeing a NULL  
pointer*/  
243     str1 = NULL;
```

01.w_Defects/function_return_value_unchecked.c:334

Level Low

Status Not processed

```
331 {  
332     char buf[100] = "";  
333     char * buf1 = buf;  
  
334     sprintf(buf1,100-strlen(STR)-1, "%s\n", STR); /*Tool should detect this line  
as error*/ /*ERROR:Return value of function never checked*/  
  
335 }  
336 }  
337
```

01.w_Defects/func_pointer.c:97

Level Low

Status Not processed

```
94 char * str_rev = NULL;  
95 if (str1 != NULL)  
96 {  
  
97     i = strlen(str1);
```

```
98     str_rev = (char *) malloc(i+1);
99     for (j = 0; j < i; j++)
100    {
```

01.w_Defects/func_pointer.c:391

Level Low**Status** Not processed

```
388 char * str_rev = NULL;
389 if (str1 != NULL)
390 {
391     i = strlen(str1);
392     str_rev = (char *) malloc(i+1);
393     for (j = 0; j < i; j++)
394     {
```

01.w_Defects/invalid_memory_access.c:505

Level Low**Status** Not processed

```
502 char * str_rev = NULL;
503 if (str1 != NULL)
504 {
505     i = strlen(str1);
506     str_rev = (char *) malloc(i+1);
507     if (str_rev != NULL)
508     {
```

01.w_Defects/main.c:22

Level Low**Status** Not processed

```
19 if(argv[1])
20 {
21
22 vflag_copy = atoi(argv[1]);
23 vflag_file = (int)floor((double)vflag_copy/1000.0);
24 vflag = (int)floor((int)vflag_copy%1000);
25 printf("vflag_file = %d vflag_func = %d vflag_copy =%d \n" , vflag_file, vflag,
vflag_copy);
```

01.w_Defects/memory_leak.c:72

Level Low

Status Not processed

```
69 {
70     char *str = "This is a string";
71     char *str1;
72     memory_leak_003_func_001(strlen(str),&str1);/*Tool should detect this line as
error*/ /*ERROR:Memory Leakage */
73     strcpy(str1,str);
74 }
75
```

01.w_Defects/memory_leak.c:399

Level Low

Status Not processed

```
396 void memory_leak_0015 ()
397 {
398     char *str = "This is a string";
399     char *str1 = memory_leak_0015_func_001(strlen(str)); /*Tool should detect this line
as error*/ /*ERROR:Memory Leakage */
400     if(str1!=NULL)
```

```
401 {  
402     strcpy(str1,str);
```

01.w_Defects/memory_leak.c:423

Level Low**Status** Not processed

```
420 void memory_leak_0016 ()  
421 {  
422     char *str = "This is a string";  
  
423     memory_leak_0016_func_001(strlen(str));  
  
424     strcpy(memory_leak_0016_gbl_ptr,str);  
425 }  
426
```

01.w_Defects/null_pointer.c:237

Level Low**Status** Not processed

```
234 void null_pointer_015 ()  
235 {  
236     char *str = "This is a string";  
  
237     null_pointer_015_func_001(strlen(str));  
  
238     strcpy(null_pointer_015_gbl_ptr,str); /*Tool should detect this line as error*/  
     /*ERROR:NULL pointer dereference*/  
239     free(null_pointer_015_gbl_ptr);  
240     null_pointer_015_gbl_ptr = NULL;
```

01.w_Defects/st_underrun.c:24

Level Low**Status** Not processed

```
21 {  
22     char buf[10];  
23     strcpy(buf, "my string");  
  
24     int len = strlen(buf) - 1;  
  
25     while (buf[len] != 'Z')  
26     {  
27         len--; /*Tool should detect this line as error*/ /* Stack Under RUN error */
```

01.w_Defects/st_underrun.c:50

Level Low

Status Not processed

```
47 void st_underrun_002_func_001 (st_underrun_002_s_001 s)  
48 {  
49  
  
50     int len = strlen(s.buf) - 1;  
  
51     for (;s.buf[len] != 'Z';len--)/*Tool should detect this line as error*/ /* Stack Under  
RUN error */  
52     {  
53         /* if (s.buf[len] == '\0')
```

01.w_Defects/st_underrun.c:85

Level Low

Status Not processed

```
82  
83 void st_underrun_003_func_002 (st_underrun_003_s_001 *s)  
84 {  
  
85     int len = strlen(s->buf) - 1;  
  
86     do  
87     {  
88         s->buf[len] = 'A';
```

01.w_Defects/st_underrun.c:122

Level Low**Status** Not processed

```
119 {  
120   st_underrun_004_s_001 s1;  
121   st_underrun_004_func_002(s);  
  
122   int len = strlen(s->buf) - 1;  
  
123   do  
124   {  
125       s->buf[len] = 'B';
```

01.w_Defects/st_underrun.c:191

Level Low**Status** Not processed

```
188 void st_underrun_006_func_001 (st_underrun_006_s_001 s)  
189 {  
190  
  
191   int len = strlen(s.buf) - 1;  
  
192   char c;  
193   for (;s.buf[len] != 'Z';len--) /*Tool should detect this line as error*/ /* Error: Stack  
Under RUN error */  
194   {
```

01.w_Defects/st_underrun.c:225

Level Low**Status** Not processed

```
222
223 void st_underrun_007_func_001 (st_underrun_007_s_001 *s)
224 {

225     int len = strlen(s->buf) - 1;

226     char c;
227     for (;s->buf[len] != 'Z';len--)
228     {
```

02.wo_Defects/buffer_underrun_dynamic.c:647

Level Low

Status Not processed

```
644 char ch='o';
645 if(destbuf!=NULL)
646 {

647     for(i=0;i<strlen(srcbuf);i++)

648     {
649         if(srcbuf[i]==ch)/*Tool should not detect this line as error*/ /*No ERROR:Buffer
Underrun*/
650     }
```

02.wo_Defects/deletion_of_data_structure_sentinel.c:42

Level Low

Status Not processed

```
39 {
40     char str1[]{"This is a string"};
41     char str[17];

42     memcpy(str,str1,strlen(str1)+1);/*Tool should not detect this line as error*/ /*No
ERROR:Deletion of a data structure sentinel*/

43 }
44
```

```
45 /*
```

02.wo_Defects/free_null_pointer.c:118

Level Low

Status Not processed

```
115 void free_null_pointer_005 ()
```

```
116 {
```

```
117   char *str = "This is a string";
```

```
118   free_null_pointer_005_func_001(strlen(str));
```

```
119   strcpy(free_null_pointer_005_gbl_ptr,str);
```

```
120   free(free_null_pointer_005_gbl_ptr);/* Tool should not detect this line as error */
```

```
/*No ERROR:Freeing a NULL pointer*/
```

```
121   free_null_pointer_005_gbl_ptr = NULL;
```

02.wo_Defects/free_null_pointer.c:248

Level Low

Status Not processed

```
245 {
```

```
246   char *str = "This is a string";
```

```
247   char *str1=NULL;
```

```
248   free_null_pointer_008_func_001(strlen(str),&str1);
```

```
249   strcpy(str1,str);
```

```
250   free(str1);/* Tool should not detect this line as error */ /*No ERROR:Freeing a NULL  
pointer*/
```

```
251   str1 = NULL;
```

02.wo_Defects/function_return_value_unchecked.c:335

Level Low

Status Not processed

```
332 {  
333     char buf[100] = "";  
334     char * buf1 = buf;  
  
335     if (snprintf(buf1,100-strlen(STR)-1, "%s\n", STR)>0 ) /*Tool should not detect this  
line as error*/ /*No ERROR:Return value of function never checked*/  
  
336     {  
337         ;  
338     }
```

02.wo_Defects/func_pointer.c:104

Level Low

Status Not processed

```
101 char * str_rev = NULL;  
102 if (str1 != NULL)  
103 {  
  
104     i = strlen(str1);  
  
105     str_rev = (char *) malloc(i+1);  
106     if(str_rev!=NULL)  
107     {
```

02.wo_Defects/func_pointer.c:407

Level Low

Status Not processed

```
404 char * str_rev = NULL;  
405 if (str1 != NULL)  
406 {  
  
407     i = strlen(str1);  
  
408     str_rev = (char *) malloc(i+1);  
409     if(str_rev !=NULL)
```

02.wo_Defects/invalid_memory_access.c:510

Level Low**Status** Not processed

```
507 char * str_rev = NULL;  
508 if (str1 != NULL)  
509 {
```

```
510     i = strlen(str1);
```

```
511     str_rev = (char *) malloc(i+1);  
512     if (str_rev != NULL)  
513     {
```

02.wo_Defects/main.c:22

Level Low**Status** Not processed

```
19 if(argv[1])  
20 {  
21
```

```
22 vflag_copy = atoi(argv[1]);
```

```
23 vflag_file = (int)floor((double)vflag_copy/1000.0);  
24 vflag = (int)floor((int)vflag_copy%1000);  
25 printf("vflag_file = %d vflag_func = %d vflag_copy =%d \n" , vflag_file, vflag,  
vflag_copy);
```

02.wo_Defects/memory_allocation_failure.c:224

Level Low**Status** Not processed

```
221 int j;
222 if (str1 != NULL)
223 {
224     static_var = strlen(str1);
225     memory_allocation_failure_007_str_gbl = (char *) malloc(static_var+1); /*Tool
should not detect this line as error*/ /*No ERROR:Memory allocation failure */
226     if(memory_allocation_failure_007_str_gbl!=NULL)
227 }
```

02.wo_Defects/memory_leak.c:75

Level Low**Status** Not processed

```
72 {
73     char *str = "This is a string";
74     char *str1;
75     memory_leak_003_func_001(strlen(str),&str1); /*Tool should not detect this line as
error*/ /*No ERROR:Memory Leakage */
76     strcpy(str1,str);
77     free(str1);
78 }
```

02.wo_Defects/memory_leak.c:405

Level Low**Status** Not processed

```
402 void memory_leak_0015 ()
403 {
404     char *str = "This is a string";
405     char *str1 = memory_leak_0015_func_001(strlen(str)); /*Tool should not detect this
line as error*/ /*No ERROR:Memory Leakage */
406     if(str1!=NULL)
```

```
407 {  
408   strcpy(str1,str);
```

02.wo_Defects/memory_leak.c:430

Level Low**Status** Not processed

```
427 void memory_leak_0016 ()  
428 {  
429   char *str = "This is a string";  
  
430   memory_leak_0016_func_001(strlen(str));  
  
431   strcpy(memory_leak_0016_gbl_ptr,str);  
432   free(memory_leak_0016_gbl_ptr);  
433   memory_leak_0016_gbl_ptr = NULL;
```

02.wo_Defects/null_pointer.c:258

Level Low**Status** Not processed

```
255 void null_pointer_015 ()  
256 {  
257   char *str = "This is a string";  
  
258   null_pointer_015_func_001(strlen(str));  
  
259   strcpy(null_pointer_015_gbl_ptr,str); /*Tool should not detect this line as error*/  
/*NO ERROR:NULL pointer dereference*/  
260   free(null_pointer_015_gbl_ptr);  
261   null_pointer_015_gbl_ptr = NULL;
```

02.wo_Defects/st_underrun.c:25

Level Low**Status** Not processed

```
22 {  
23     char buf[10];  
24     strcpy(buf, "my string");  
  
25     int len = strlen(buf) -1;  
  
26     while (buf[len] != 'Z')  
27     {  
28         len--; /*Tool should not detect this line as error*/ /* No Stack Under RUN  
error */
```

02.wo_Defects/st_underrun.c:51

Level Low**Status** Not processed

```
48 void st_underrun_002_func_001 (st_underrun_002_s_001 s)  
49 {  
50  
  
51     int len = strlen(s.buf) - 1;  
  
52     for (;s.buf[len] != 'Z';len--) /*Tool should not detect this line as error*/ /* No Stack  
Under RUN error */  
53     {  
54         if ( len < 0 )
```

02.wo_Defects/st_underrun.c:86

Level Low**Status** Not processed

```
83  
84 void st_underrun_003_func_002 (st_underrun_003_s_001 *s)  
85 {  
  
86     int len = strlen(s->buf) - 1;  
  
87     do  
88     {
```

```
89         s->buf[len] = 'A';
```

02.wo_Defects/st_underrun.c:123

Level Low

Status Not processed

```
120 {  
121   st_underrun_004_s_001 s1;  
122   st_underrun_004_func_002(s);
```

```
123   int len = strlen(s->buf) - 1;
```

```
124   do  
125   {  
126       s->buf[len] = 'B';
```

02.wo_Defects/st_underrun.c:234

Level Low

Status Not processed

```
231  
232 void st_underrun_007_func_001 (st_underrun_007_s_001 *s)  
233 {
```

```
234   int len = strlen(s->buf) - 1;
```

```
235   char c = 0;  
236   for (;s->buf[len] != 'Z';len--)  
237   {
```

02.wo_Defects/wrong_arguments_func_pointer.c:380

Level Low

Status Not processed

```
377 char *str1;
```

```
378 void (*fptr)(int,char **);  
379 fptr = wrong_arguments_func_pointer_013_func_001;
```

380 fptr(strlen(str),&str1); /*Tool should not detect this line as error///*No ERROR:Wrong arguments passed to a function pointer*/

```
381 strcpy(str1,str);  
382 free(str1);  
383 str1 = NULL;
```

03.w_Defects_Cpp/main.cpp:24

Level Low

Status Not processed

```
21 {  
22     if(argv[1])  
23     {  
  
24         vflag_copy = atoi(argv[1]);  
  
25         vflag_file = (int)floor((double)vflag_copy/1000.0);  
26         vflag = (int)floor((int)vflag_copy%1000);  
27         printf("vflag_file = %d vflag_func = %d vflag_copy =%d \n" , vflag_file, vflag,  
vflag_copy);
```

04.wo_Defects_Cpp/main.cpp:23

Level Low

Status Not processed

```
20 {  
21     if(argv[1])  
22     {  
  
23         vflag_copy = atoi(argv[1]);  
  
24         vflag_file = (int)floor((double)vflag_copy/1000.0);  
25         vflag = (int)floor((int)vflag_copy%1000);  
26         printf("vflag_file = %d vflag_func = %d vflag_copy =%d \n" , vflag_file, vflag,
```

```
vflag_copy);
```

Scan Settings

1/1 02/26/2020 17:00:46

Source code <https://github.com/mmacala/itc-benchmarks-test.git>

Exclude from Analysis —

Branch in Repository master

Languages

- | | | | | |
|--|--|--|--|--|
| <input checked="" type="checkbox"/> ABAP | <input checked="" type="checkbox"/> Go | <input checked="" type="checkbox"/> PHP | <input checked="" type="checkbox"/> Swift | <input checked="" type="checkbox"/> Visual Basic 6 |
| <input checked="" type="checkbox"/> Apex | <input checked="" type="checkbox"/> Groovy | <input checked="" type="checkbox"/> PL/SQL | <input checked="" type="checkbox"/> T-SQL | <input checked="" type="checkbox"/> Vyper |
| <input checked="" type="checkbox"/> C# | <input checked="" type="checkbox"/> HTML5 | <input checked="" type="checkbox"/> Python | <input checked="" type="checkbox"/> TypeScript | <input checked="" type="checkbox"/> 1C |
| <input checked="" type="checkbox"/> C/C++ | <input checked="" type="checkbox"/> Java, Scala, | <input checked="" type="checkbox"/> Perl | <input checked="" type="checkbox"/> VB.NET | |
| <input checked="" type="checkbox"/> COBOL | <input checked="" type="checkbox"/> JavaScript | <input checked="" type="checkbox"/> Ruby | <input checked="" type="checkbox"/> VBA | |
| <input checked="" type="checkbox"/> Delphi | <input checked="" type="checkbox"/> Objective-C | <input checked="" type="checkbox"/> Solidity | <input checked="" type="checkbox"/> VBScript | |

Java/Scala/Kotlin Specific Settings

Do not build project (project is already built)

C/C++ Specific Settings

Visual Studio Project

Python Specific Settings

Python 2 Python 3

JavaScript Specific Settings

Analyze standard libraries

General Analysis Settings

Analyze libraries and nested archives

Use extra rules

Analyze config files

Incremental analysis

Source Code Charset UTF-8

Filename Charset UTF-8

Rule Sets

Export Settings

Project Information

- Security Level Dynamics
- Vulnerability Dynamics

Scan History

- Do not export scan history
- Export entire scan history
- Export the latest scans 0

Vulnerability Classification

By severity

Scan Information

- Detected vulnerabilities chart
- Vulnerability type chart
- Language statistics
- Scan error information
- Scan Settings

Issues Filter

Severity Level

- Critical
- Medium
- Low

Vulnerability Types

- Vulnerabilities in standard Java, Scala, Kotlin libraries
- Vulnerabilities in .class files that could not be decompiled
- With a task created in Jira

- Vulnerabilities without recommendations for setting up a WAF

Languages

<input checked="" type="checkbox"/> ABAP	<input checked="" type="checkbox"/> Delphi	<input checked="" type="checkbox"/> Kotlin	<input checked="" type="checkbox"/> Ruby	<input checked="" type="checkbox"/> VB.NET
<input checked="" type="checkbox"/> Android	<input checked="" type="checkbox"/> Go	<input checked="" type="checkbox"/> Objective-C	<input checked="" type="checkbox"/> Scala	<input checked="" type="checkbox"/> VBA
<input checked="" type="checkbox"/> Apex	<input checked="" type="checkbox"/> Groovy	<input checked="" type="checkbox"/> PHP	<input checked="" type="checkbox"/> Solidity	<input checked="" type="checkbox"/> VBScript
<input checked="" type="checkbox"/> C#	<input checked="" type="checkbox"/> HTML5	<input checked="" type="checkbox"/> PL/SQL	<input checked="" type="checkbox"/> Swift	<input checked="" type="checkbox"/> Visual Basic 6
<input checked="" type="checkbox"/> C/C++	<input checked="" type="checkbox"/> Java	<input checked="" type="checkbox"/> Python	<input checked="" type="checkbox"/> T-SQL	<input checked="" type="checkbox"/> Vyper
<input checked="" type="checkbox"/> COBOL	<input checked="" type="checkbox"/> JavaScript	<input checked="" type="checkbox"/> Perl	<input checked="" type="checkbox"/> TypeScript	<input checked="" type="checkbox"/> 1C

Vulnerability Table

Vulnerability Statuses

- Not processed
- Confirmed
- Rejected

Vulnerability Occurrences List

- Do not export
- Export all entries
- Export no more than entries 0

Detailed Results

Vulnerability Statuses

- Not processed
- Confirmed
- Rejected

Vulnerability Occurrences List

- Do not export

- Export all entries
- Export no more than entries 0

Source code

- Do not export source code
- Export entire vulnerable source code file
- Export context in the number of lines of code 3

Trace

- Do not export trace items
 - Export only the first and last items
 - Export all items
-
- Vulnerability comment
 - Jira information

WAF Setting Up Recommendations

Recommendations for vulnerability statuses

- Not processed
- Confirmed
- Rejected

Settings for WAF

- Imperva SecureSphere
- ModSecurity
- F5

General Report Settings

- Report Export Settings

Table of Contents