PA193: Project – Static Checking of Miranda (http://www.miranda-ng.org/en/)

Miranda NG (Next Generation) is a multi-protocol instant messaging client for Windows. It provides a basic client framework, GUI, and an advanced plugin architecture.

Tools Used: CppCheck, PREfast

Source Code Summary

```
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C++                            217          13229           6584          70101
C/C++ Header                    73           1218           1703           5718
Windows Resource File           18            436            395           3448
Windows Module Definition        2              2              0            284
-------------------------------------------------------------------------------
SUM:                           310          14885           8682          79551
-------------------------------------------------------------------------------
```

CppCheck Results Summary

S.No.	Severity	Types & number of problems reported	
1	Error	Possible null pointer dereference	5
		Memory leak	1
		Common realloc mistake: 'p' nulled but not freed upon failure	1
		Null pointer dereference	3
		Inconclusive Possible null pointer dereference	12
		Dangerous usage of 'buf' (strncpy doesn't always NULL terminate it)	1
		Uninitialized variable	1
			24
2	Warning	Member variable is not initialized in the constructor	86
		Inconclusive Division with signed and unsigned operators	2
			88

PREfast Results Summary

No.	Description	
C6001	Using uninitialized memory	5
C6011	Dereferencing null pointer	17
C6031	Return value ignored	19
C6053	Zero termination from call	6
C6054	Zero termination missing	3
C6102	Using variable from failed function call	11
C6255	Unprotected use of alloca	6
C6262	Excessive stack usage	7

C6263	Using alloca in loop	1
C6269	Pointer dereference ignored	7
C6282	Assignment replaces test	2
C6308	Realloc leak	7
C6385	Read overrun	30
C6386	Write overrun	2
C6387	Invalid parameter value	30
C28125	The function must be called from within a try/except block	1
C28159	Consider using another function instead	1
C28182	Dereferencing a copy of a null pointer	5
C28183	The argument could be one value, and is a copy of the value found in	1
C28251	Inconsistent annotation for function	1
	Total	**162**

MS Visual Studio analyzer (recommended native rules) (by category)

Ambiguous intent	75
Annotation syntax	16
Concurrency	46
Deprecated API	331
Exception handling	496
Incorrect API use	660
Kernel mode	117
Memory safety	698
Memory usage	216
mspft	327
Security	366
Type mismatch	12
Total	**3360**

Report analysis

After manual evaluation of a sample of reported issues, it is estimated that 5—15% of the warnings point to legitimate mistakes, and further 20—40% of warnings point to potential problems that are benign unless the surrounding code is changed. Almost all of the remaining warnings are false positives that could be silenced by redundant runtime checks.

Out of the legitimate mistakes, most are accounted for by missing checks of return values of library functions, notably in dynamic memory management, where the problem would only present itself in out-of-memory conditions. The second biggest culprit are functions working with wide character strings, where parameter to the function is the number of characters, but the number of bytes is passed as an argument, resulting in potential buffer overflows.

During manual evaluation, 15 legitimate problems have been fixed and several false positives have been silenced. Changes are pending review from developers.