Rule	Description	Severity	Likelihood	Remediation Cost	Priority
CRT52	Guarantee that library functions do not overflow	High	Likely	Medium	P18
CTR55	Do not use an additive operator on an iterator if the result would overflow	High	Likely	Medium	P18
STR50	Guarantee that storage for strings has sufficient space for character data and the null terminator	High	Likely	Medium	P18
STR51	Do not attempt to create a std::string from a null pointer	High	Likely	Medium	P18
MEM50	Do not access freed memory	High	Likely	Medium	P18
MEM51	Properly deallocate dynamically allocated resources	High	Likely	Medium	P18
MEM52	Detect and handle memory allocation errors	High	Likely	Medium	P18
MEM53	Explicitly construct and destruct objects when manually managing object lifetime	High	Likely	Medium	P18
MEM54	Provide placement new with properly aligned pointers to sufficient storage capacity	High	Likely	Medium	P18
MEM55	Honor replacement dynamic storage management requirements	High	Likely	Medium	P18
MEM56	Do not store an already-owned pointer value in an unrelated smart pointer	High	Likely	Medium	P18
ERR54	Catch handlers should order their parameter types from most derived to least derived	Medium	Likely	Low	P18
MSC51	Ensure your random number generator is properly seeded	High	Likely	Medium	P18
EXP60	Do not pass a nonstandard-layout type object across execution boundaries	High	Probable	Medium	P12
EXP53	Do not read uninitialized memory	High	Probable	Medium	P12
DCL50	Do not define a C-style varadic function	High	Probable	Medium	P12