Project	S.No	Leak ID	Buggy Parent ID	Commit ID	Alloc line no.	Free line no.	Notes	Result	Github commit verified for a Memory Leak?	Github verified fix location compared to our fix location		Memfix fix					
Previous fixed bugs:											Criteria for Result to be a Success for AddressWatcher				Memfix		
binutils	1	1	a506516	be74fad95edc8827516e144cf38d135b503249cd	178,184	381	malloced pointer returned from function to copy_main at end of which pointer tmpname not freed	Success	Yes	Yes		No output			12 fail to output	binutils_leak_1.c:Status: Failed binutils_leak_7.c:Status: Failed binutils_leak_8.c:Status: Failed git_leak_10.c:Status: Failed	
	2	2	2c244f9	3cfd3dd0956fe854a07795de12c1302ecabbd819	218	240	in function write_archive the pointer new_name is malloced but not freed at end of function. This is a simple case for address watcher	Success	Yes	Yes		Solved					
		3															
		4	52a93b9	a26a013f22a19e2c16729e64f40ef8a7dfcc086e	74	180	In function line_info_table in fail path table->sequences is not freed. Address sanitizer cannot help here	Failure	Yes	Yes		No output					
	3	5	e13cb30	7ed1acafa0b5d135342f9dcc0eb0387dff95005a	52,60	39	Code organisation choice in this case does all free operations in btrace_clear. So since there is no read of leaked variable in btrace_clear, addresswatcher does not help in this case.However Addresswatcher tags two lines before btrace_clear in main saving developer time.	Failure	Yes	Yes		No output					
	4	6	2f5404b	f978cb06dbfbd93dbd52bd39d992f8644b0c639e	40.66	143	The variable sect_opts is malloced at start of add_symbol_file_command but not freed at end of it. The last use of sect_opt is within a for loop so the free will have to be inserted after the for loop. Still valuable information can be gleaned from address watcher	Failure	Yes	NA (The variable was not malloced in the fix)		No output					
	5	7	ad36c6c	3f2a3564b1c3872e4a380f2484d40ce2495a4835	43	109	malloced pointer returned to function do_start_initialization as arg to concat but not freed	Success	Yes	Yes		No output					
	6	8	c42608e	848ac659685fba46ce8816400db705f60c8040f7	23	152.160,174	In function copy_section in error paths the pointer memhunk is not freed. But luckily in this case within error condition we do check memhunk. So Addresswatcher can help here	Success	Yes	Yes		Failed					
	7	9	9d2cdc8	aba19b625f34fb3d61263fe8044cf0c6d8804570	53	286	The function ada_demangle parses an input string according to a format. When it cannot continue parsing by the format it jumps to an 'unknown' label where the pointer demangle is not freed. Address watcher could not see it because on error path there is no load/store of this chunk.	Failure	Yes	Yes		Solved	1				
tmux	8	1	7ba5ad4	c363c236aaea5b7a879493d8f3c85bead546f063	224	229	In function screen_redraw_make_pane_status the pointer out is malloced but not freed. This is a straightforward case where Addresssanitizer works perfectly	Success	Yes			Solved					
	9	2	ae1a6c2	1e0eb914d945e0f287716d56669d0de409e86e59	149,151	155	In function status_prompt_complete after xasprintf the pointer pointing to allocated memory is reassigned before this it should be freed. Address ssanitizer can detct this because of the xasprintf	Success	Yes	Yes		Solved					
	10	3	c8ecbf3	2c9bdd9e326723fb392aed4d8df12cba7ef34f1f	77,81	128,168	cmd_load_buffer_exec calls a function that mallocs a pointer, on error case it abrupty does goto error midexecution where it is not freed. Addresswatcher shows only the use of memory. before jumping to error Label. There is a similiar problem for cmd_save_buffer_exec but there is a read to malloced memory in error condition so we are able to point correct location here.	One free failed, second success	Yes	Yes		Solved	7				
	11	4	54bcaab	d566c780e54010112d499707cd80a594144d1a89	46	108	Pointer base is malloced and returned to the function make_label but not freed in end	Success	Yes	Yes		Solved	2				
	12	5	40fefe2	933929cd622478bb43afe590670613da2e9ff359	176	187,192,195	In function cmd_if_shell_exec there are 3 rees to be inserted on 3 error paths. Addresssanitizer correctly identifies 1 of them because on one error path the if condition uses malloced memory	One free success	Yes			Solved					
	13	6	695a591	7340d5adfdc8cc6d845a373f3e0d59bfd10a45d1	285,289,314	349	In function cmd_capture_pane_exec in error path the variable buf is allocated but not freed	Failure	Yes			No output					
	14	7	540f0b3	189017c078b7870c18ced485c1fd99f65fcc4801	184,192	249	Last use within an if condition in function cmd_retval. We need to put last use outside if closing block. Address sanitizer cannot help in this case.	Failure	Yes	Yes		No output					
	15	8		5acee1c04ed38afd6a32da4a66e6855ccdc52af3	115	111	In function get_proc_name on error condition the pointer buf is not freed. AddressWatcher cannot catch this because there is no read/write to this memory inside error condition. Address Watcher cannot help	Failure	Yes	Yes(used goto error where it is freed instead o		No output					
	16	9	69b7c49	6daf06b1ad61f67e9f7780d787451b9b5f82dd43	173	183	In function cmd_load_buffer_callback on error path return the pointer pdata is not freed. Similiarily on error path in paste_replace the alias data is not freed.	One free failed, s	Yes	Yes		No output					
openssh-portable	17	1	?	b2afdaf1b52231aa23d2153f4a8c5a60a694dda	187	193,216	The function match_filter_list matches two strings by some rules. If the strings could not be malloced then on this error condition an abrupt return happens. One string is not freed. Since leaked string eror condition is very close to the malloc line register passing prevents instrumentation from detecting this line. But because it is close to malloc the leak can be fixed easily without Address Watcher. In the main loop the second leaks fix is correctly identified.	Failure	Yes	Yes		Solved					
	18	2		66d2e229baa9fe57b86	40	44	Error condition return in function ssh_packet_write. Malloced memory not used before error condition return hence address watcher cannot track a last use before return.	Failure	Yes	Yes		Solved					
	19	3	e6b9503	a63cfa26864b93ab6afefad0b630e5358ed8edfa	147	150	In function client_input_host_keys when argc is 1 abrupt termination happens without freeing ctx	Failure	Yes	No the allocation was moved below abrupt termination		No output					
	20	4	7ad8b28	4f7cc2f8cc861a21e6dbd7f6c25652afb38b9b96	65	65	In function main there is a goto loop where logfile is repe	Failure	Yes	No the allocation was replaced with pointer reassignment							
	21	5	?	64a89ec07660abba4d0da7c0095b7371c98bab62	192	202	In function do_init object ret is allocated but on error path it is not freed. Since inside error path this object is not read/written it cannot help	Failure	Yes	Yes		Solved					

Project	S.No	Leak ID	Buggy Parent ID	Commit ID	Alloc line no.	Free line no.	Notes	Result	Github commit verified for a Memory Leak?	Github verified fix location compared to our fix location		Memfix fix					
	22	6		165bc8786299e261706ed60342985f9de93a7461			In function ssh_ed25519_verify in error path variables b	Failure	Yes	Yes		Solved					
	23	7	?	[redacted]	87	138	In function main sig->r and sig->s is dereferenced.	Success	Yes	NA since they removed the reassignment in the firs place		No output (recursive functions)					
	24	8	0cce17f	e52a260f16888ca75390f97de4606943e61785e8	32	151	In function load_identity_file when perm_ok is false abrupt stop to execution. However pointer private not free before this. Again there is no use before this so AddressWatcher cannot help	Failure	Yes	Yes		No output					
	25	9	534b2cc	393920745fd328d3fe07f739a3cf7e1e6db45b60	179	254	In function main do_readdir is called which allocates dir_entries but due to some other error it returns -1. In this error path it must free dir_entries. Since this if condition directly allocates this memory we are able to track it	Success	Yes	No code reorganized		No output					
	26	10	?	[redacted]	45	108	The function update_krl_from_file expands a given filename. For the expanded filenam memory is allocated fro heap but not freed at exit of function.	Success	Yes	Yes		Solved					
to be fixed																	
linux	27		TO BE MERGED	TO BE MERGED	223	248	In arch/x86/entry/vdso/vdso2c.c a variable name is allocated but not freed.at end of main function	Success									