Some of these hooks are populated with initialisation values which are cleared after the first call to a function. For example __malloc_hook is populated with the address of the malloc_hook_ini() function during GLIBC initialisation, which zeroes __malloc_hook and calls ptmalloc_init().

```
static void* malloc_hook_ini (size_t sz, const void* caller) {
	__malloc_hook = NULL;
	ptmalloc_init();
	return __libc_malloc(sz);
}
```

When a hook is zeroed, calls to its parent function go straight through to that function. When a hook is populated, execution is redirected to the address pointed to by the hook when the parent function is called. For example, the first lines of __libc_malloc() are as follows:

```
void* (*hook) (size_t, const void*) = atomic_forced_read(__malloc_hook);
if(__builtin_expect(hook != NULL, 0))
	return(*hook)(bytes, RETURN_ADDRESS(0));
```

## Mitigations

Historic exploit mitigations introduced into GLIBC malloc.

| Commit date | Published in GLIBC version | Author | Description | Diff |
|---|---|---|---|---|
| 19/08/2003 | 2.3.3 | Ulrich Drepper | Ensure chunks don't wrap around memory on free(). | diff |
| 21/08/2004 | 2.3.4 | Ulrich Drepper | Safe unlinking checks. | diff |
| 09/09/2004 | 2.3.4 | Ulrich Drepper | Check that the chunk being freed is not the top chunk. Check the next chunk on free is not beyond the bounds of the heap. Check that the next chunk has its prev_inuse bit set before free. | diff |
| 19/11/2004 | 2.3.4 | Ulrich Drepper | Check next chunk's size sanity on free(). | diff |
| 20/11/2004 | 2.3.4 | Ulrich Drepper | Check chunk about to be returned from fastbin is the correct size. Check that the chunk about to be returned from the unsorted bin has a sane size. | diff |
| 22/12/2004 | 2.3.4 | Ulrich Drepper | Ensure a chunk is aligned on free(). | diff |
| 13/10/2005 | 2.4 | Ulrich Drepper | Check chunk is at least MINSIZE bytes on free(). | diff |
| 30/04/2007 | 2.6 | Ulrich Drepper | Unsafe unlink checks for largebins. | diff |
| 19/06/2009 | 2.11 | Ulrich Drepper | Check if bck->fd != victim when allocating from a smallbin. Check if fwd->bk != bck before | diff |

| | | | | |
|---|---|---|---|---|
| | | | adding a chunk to the unsorted bin whilst remaindering an allocation from a large bin. Check if fwd->bk != bck before adding a chunk to the unsorted bin whilst remaindering an allocation from a binmap search. Check if fwd->bk != bck when freeing a chunk directly into the unsorted bin. | |
| 03/04/2010 | 2.12 | Ulrich Drepper | When freeing a chunk directly into a fastbin, check that the chunk at the top of the fastbin is the correct size for that bin. | diff |
| 17/03/2017 | 2.26 | DJ Delorie | Size vs prev_size check in unlink macro. | diff |
| 30/08/2017 | 2.27 | Florian Weimer | Don't backtrace on abort anymore. | diff |
| 30/11/2017 | 2.27 | Arjun Shankar | Fix integer overflow when allocating from the tcache. | diff |
| 12/01/2018 | 2.27 | Istvan Kurucsai | Fastbin size check in malloc_consolidate. | diff |
| 14/04/2018 | 2.28 | DJ Delorie | Check if bck->fd != victim when removing a chunk from the unsorted bin during unsorted bin iteration. | diff |
| 16/08/2018 | 2.29 | Pochang Chen | Check top chunk size field sanity in use_top. | diff |
| 17/08/2018 | 2.29 | Moritz Eckert | Proper size vs prev_size check before unlink() in backward consolidation via free. Same check in malloc_consolidate(). | diff |
| 17/08/2018 | 2.29 | Istvan Kurucsai | When iterating unsorted bin check: size sanity of next chunk on heap to removed chunk, next chunk on heap prev_size matches size of chunk being removed, check bck->fd != victim and victim->fd != unsorted_chunks (av) for chunk being removed, check prev_inuse is not set on next chunk on heap to chunk being removed. | diff |
| 20/11/2018 | 2.29 | DJ Delorie | Tcache double-free check. | diff |
| 26/11/2018 | 2.29 | Florian Weimer | Validate tc_idx before checking for tcache double-frees. | diff |
| 14/03/2019 | 2.30 | Adam Maris | Check for largebin list corruption when sorting into a largebin. | diff |
| 18/04/2019 | 2.30 | Adhemerval Zanella | Request sizes cannot exceed PTRDIFF_MAX (0x7fffffffffffffff) | diff |

## Mitigation error messages

GLIBC malloc error messages and their corresponding mitigations. Not an exhaustive list, but the most frequently triggered mitigations with regards to exploit development.

| Error message | Triggered mitigation |
|---|---|
| "double free or corruption (fasttop)" | Fastbins double-free check. |
| "malloc(): memory corruption (fast)" | Size field check during allocation from fastbins. |
| "corrupted double-linked list" | Safe unlinking check in the unlink macro/function. |
| "corrupted size vs. prev_size" | Size vs. prev_size check in the unlink macro/function. |
| "corrupted size vs. prev_size while consolidating" | Size vs. prev_size check (pre-unlink). |
| "corrupted double-linked list (not small)" | Largebins nextsize safe unlinking check in the unlink macro/function. |
| "malloc(): smallbin double linked list corrupted" | Link integrity check during allocation from smallbins. |
| "malloc(): invalid size (unsorted)" | Size field integrity check during allocation from unsortedbin. |
| "malloc(): invalid next size (unsorted)" | Next size check during allocation from unsortedbin. |
| "malloc(): mismatching next->prev_size (unsorted)" | prev_size check during allocation from unsortedbin. |
| "malloc(): unsorted double linked list corrupted" | fd link integrity check during allocation from unsortedbin. |
| "malloc(): invalid next->prev_inuse (unsorted)" | prev_inuse check during allocation from unsortedbin. |
| "malloc(): corrupted unsorted chunks 3" | bk link integrity check during allocation from unsortedbin. |
| "malloc(): largebin double linked list corrupted (nextsize)" | Nextsize link integrity check during allocation from largebins. |
| "malloc(): largebin double linked list corrupted (bk)" | bk link integrity check during allocation from largebins. |
| "malloc(): corrupted unsorted chunks" | Link integrity check during remaindering from largebins. |
| "malloc(): corrupted unsorted chunks 2" | Link integrity check during remaindering from a binmap search. |
| "malloc(): corrupted top size" | Top chunk size integrity check during allocation from top chunk. |

| | |
|---|---|
| "free(): invalid pointer" | Wraparound & alignment check during free. |
| "free(): invalid size" | Set 4th bit check during free. |
| "free(): double free detected in tcache 2" | Tcache double-free check. |
| "free(): invalid next size (fast)" | Nextsize check during free (fast chunks only). |
| "double free or corruption (top)" | Top chunk free check (non-fast sizes). |
| "double free or corruption (out)" | Heap boundary check (non-fast sizes). |
| "double free or corruption (!prev)" | Non-fast double-free check. |
| "free(): invalid next size (normal)" | Nextsize check during free (non-fast sizes). |
| "free(): corrupted unsorted chunks" | Unsortedbin lnk integrity check during free. |