author	Andre Przywara <andre.przywara@arm.com>	2025-03-20 15:55:57 +0000
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2025-05-02 08:01:34 +0200
commit	dba5a1f963cf781c0b60f4b7f07465a6c687c27e (patch)	
tree	977ea74521d6910019012ce1ca89832d6ef1fdf9	
parent	d65216720eab66413f66ca5a49627afebedb734c (diff)	
download	linux-dba5a1f963cf781c0b60f4b7f07465a6c687c27e.tar.gz	

cpufreq: sun50i: prevent out-of-bounds access

```
[ Upstream commit 14c8a418159e541d70dbf8fc71225d1623beaf0f ]

A KASAN enabled kernel reports an out-of-bounds access when handling the
nvmem cell in the sun50i cpufreq driver:
================================================================
BUG: KASAN: slab-out-of-bounds in sun50i_cpufreq_nvmem_probe+0x180/0x3d4
Read of size 4 at addr ffff000006bf31e0 by task kworker/u16:1/38

This is because the DT specifies the nvmem cell as covering only two
bytes, but we use a u32 pointer to read the value. DTs for other SoCs
indeed specify 4 bytes, so we cannot just shorten the variable to a u16.

Fortunately nvmem_cell_read() allows to return the length of the nvmem
cell, in bytes, so we can use that information to only access the valid
portion of the data.
To cover multiple cell sizes, use memcpy() to copy the information into a
zeroed u32 buffer, then also make sure we always read the data in little
endian fashion, as this is how the data is stored in the SID efuses.

Fixes: 6cc4bcceff9a ("cpufreq: sun50i: Refactor speed bin decoding")
Reported-by: Jernej Skrabec <jernej.skrabec@gmail.com>
Signed-off-by: Andre Przywara <andre.przywara@arm.com>
Reviewed-by: Jernej Škrabec <jernej.skrabec@gmail.com>
Signed-off-by: Viresh Kumar <viresh.kumar@linaro.org>
Signed-off-by: Sasha Levin <sashal@kernel.org>
```

Diffstat

-rw-r--r--	drivers/cpufreq/sun50i-cpufreq-nvmem.c	18

1 files changed, 12 insertions, 6 deletions

```
diff --git a/drivers/cpufreq/sun50i-cpufreq-nvmem.c b/drivers/cpufreq/sun50i-cpufreq-nvmem.c
index 47d6840b348994..744312a44279cb 100644
--- a/drivers/cpufreq/sun50i-cpufreq-nvmem.c
+++ b/drivers/cpufreq/sun50i-cpufreq-nvmem.c
@@ -194,7 +194,9 @@ static int sun50i_cpufreq_get_efuse(void)
        struct nvmem_cell *speedbin_nvmem;
        const struct of_device_id *match;
        struct device *cpu_dev;
-       u32 *speedbin;
+       void *speedbin_ptr;
```

```
+       u32 speedbin = 0;
+       size_t len;
        int ret;

        cpu_dev = get_cpu_device(0);
@@ -217,14 +219,18 @@ static int sun50i_cpufreq_get_efuse(void)
                return dev_err_probe(cpu_dev, PTR_ERR(speedbin_nvmem),
                                     "Could not get nvmem cell\n");

-       speedbin = nvmem_cell_read(speedbin_nvmem, NULL);
+       speedbin_ptr = nvmem_cell_read(speedbin_nvmem, &len);
        nvmem_cell_put(speedbin_nvmem);
-       if (IS_ERR(speedbin))
-               return PTR_ERR(speedbin);
+       if (IS_ERR(speedbin_ptr))
+               return PTR_ERR(speedbin_ptr);

-       ret = opp_data->efuse_xlate(*speedbin);
+       if (len <= 4)
+               memcpy(&speedbin, speedbin_ptr, len);
+       speedbin = le32_to_cpu(speedbin);

-       kfree(speedbin);
+       ret = opp_data->efuse_xlate(speedbin);
+
+       kfree(speedbin_ptr);

        return ret;
 };
```