From: Jincheng Wang <jc.w4ng@gmail.com>
To: u-boot@lists.denx.de
Subject: Out of bounds write vulnerability in the sqfs_readdir() function
Date: Thu, 26 May 2022 16:28:07 +0800 [thread overview]
Message-ID: <CALO=DHFB+yBoXxVr5KcsK0iFdg+e7ywko4-e+72kjbcS8JBfPw@mail.gmail.com> (raw)

[-- Attachment #1: Type: text/plain, Size: 1458 bytes --]

Hello u-boot list,

I found the sqfs_readdir() function is vulnerable to Out-of-Bound write,
which will cause arbitrary code execution.

```
int sqfs_readdir(struct fs_dir_stream *fs_dirs, struct fs_dirent **dentp)
{
......
/* Set entry name */

strncpy(dent->name, dirs->entry->name, dirs->entry->name_size + 1);
dent->name[dirs->entry->name_size + 1] = '\0';

offset = dirs->entry->name_size + 1 + SQFS_ENTRY_BASE_LENGTH;
dirs->entry_count--;
.......
}


struct squashfs_dir_stream {
struct fs_dir_stream fs_dirs;
struct fs_dirent dentp;
size_t size;
int entry_count;
struct squashfs_directory_header *dir_header;
struct squashfs_directory_entry *entry;
......
};


static int sqfs_search_dir(struct squashfs_dir_stream *dirs, char
**token_list,
  int token_count, u32 *m_list, int m_count)
{
......
while (!sqfs_readdir(dirsp, &dent)) {
ret = strcmp(dent->name, token_list[j]);
if (!ret)
break;
free(dirs->entry);
dirs->entry = NULL;
}
......
}
```

The sqfs_readdir() function use strncpy to set entry name, while the type
of dirs->entry->name_size is defined as "u16" in the struct
squashfs_directory_entry
and dent->name
is defined as "char[256]" in the struct fs_dirent.

We can overwrite *dirs_header and *entry in the struct squashfs_dir_stream,
so that we can use the sqfs_search_dir() function to free a fake
chunk which causes arbitrary code execution.
You can see the Poc in the attachment.

```
host bind 0 test4.sqfs
ls host 0 /dirs
```

[-- Attachment #2: test4.sqfs --]
[-- Type: application/octet-stream, Size: 4096 bytes --]

Reply instructions:

You may reply publicly to this message via plain-text email
using any one of the following methods:

* Save the following mbox file, import it into your mail client,
  and reply-to-all from there: mbox

  Avoid top-posting and favor interleaved quoting:
  https://en.wikipedia.org/wiki/Posting_style#Interleaved_style

* Reply using the **--to**, **--cc**, and **--in-reply-to**
  switches of git-send-email(1):

  ```
  git send-email \
    --in-reply-to='CALO=DHFB+yBoXxVr5KcsK0iFdg+e7ywko4-e+72kjbcS8JBfPw@mail.gmail.com' \
    --to=jc.w4ng@gmail.com \
    --cc=u-boot@lists.denx.de \
    /path/to/YOUR_REPLY
  ```

  https://kernel.org/pub/software/scm/git/docs/git-send-email.html

* If your mail client supports setting the **In-Reply-To** header
  via mailto: links, try the mailto: link

Be sure your reply has a **Subject:** header at the top and a blank line before the message body.
