New issue

Jump to bottom

# NULL pointer derefence during HTTP authentication #1242

Closed pjlantz opened this issue on Jul 25, 2020 · 3 comments · Fixed by #1243

Assignees

Labels p:critical t:bug

pjlantz commented on Jul 25, 2020 • edited

Cherokee Web Server 0.4.27 to 1.2.104 have a NULL pointer dereference which leads to a denial of service.
Any server that has HTTP authentication (either basic or digest) enabled and paths that respond with the WWW-Authenticate header, can be crashed by an unauthenticated and remote attacker by sending a malformed Authorization header to such paths.

The following commands are used to generate HTTP requests that trigger the vulnerability

```
1) curl -H "Authorization: Basic " <url>
2) curl -H "Authorization: Digest " <url>
```

`cherokee_buffer_add` does not allocate memory if the the size of the input string is less or equal to zero and return `ret_ok` nonetheless.

```
ret_t
cherokee_buffer_add (cherokee_buffer_t *buf, const char *txt, size_t size)
{
        int available;

        if (unlikely (size <= 0))
                return ret_ok;
.
.
.
```

`cherokee_validator_parse_digest` and `cherokee_validator_parse_basic` do not have any checks on the return value from `cherokee_buffer_add` and will later dereference an uninitialized pointer (read and write), at `validator.c:180`

```
ret_t
cherokee_validator_parse_digest (cherokee_validator_t *validator,
                                 char *str, cuint_t str_len)
{
        cuint_t             len;
        char               *end;
        char               *entry;
        char               *comma;
        char               *equal;
        cherokee_buffer_t   auth = CHEROKEE_BUF_INIT;
        cherokee_buffer_t  *entry_buf;

        /* Copy authentication string
         */
        cherokee_buffer_add (&auth, str, str_len);

        entry = auth.buf;
        end   = auth.buf + auth.len;

        do {
                /* Skip some chars
                 */
                while ((*entry == CHR_SP) ||
                       (*entry == CHR_CR) ||
                       (*entry == CHR_LF)) entry++;
.
.
.
```

and in a call to `cherokee_buffer_decode_base64` (illegal write at `buffer.c:1681` ) respectively

```
ret_t
cherokee_validator_parse_basic (cherokee_validator_t *validator, char *str, cuint_t str_len)
{
        char              *colon;
        cherokee_buffer_t  auth = CHEROKEE_BUF_INIT;

        /* Decode base64
         */
        cherokee_buffer_add (&auth, str, str_len);
        cherokee_buffer_decode_base64 (&auth);
.
.
.
```

skinkie self-assigned this on Jul 25, 2020

skinkie added p:critical t:bug labels on Jul 25, 2020

**skinkie** mentioned this issue on Jul 25, 2020

**Fix CVE-2020-12845** #1243

Merged

**skinkie** commented on Jul 25, 2020 — Member

The buffer here you mention here is statically initialized by CHEROKEE_BUF_INIT. The return value check would also in case of unallocatable size not be the solution. On more places in the code the return value of these calls are not checked and has been observed as "not an issue" by the original developer. The actual issue is here that it is assumed that auth.len > 0. The fix guards both downstream and upstream function, so empty input should be prevented now.

**skinkie** commented on Jul 25, 2020 — Member

**@pjlantz** would you be so kind to confirm the pull request solves your findings?

**pjlantz** commented on Jul 25, 2020 — Author

I can confirm that there is no segmentation fault occurring anymore

**skinkie** closed this as completed in #1243 on Jul 25, 2020

Assignees

**skinkie**

Labels

p:critical t:bug

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

**Fix CVE-2020-12845**
cherokee/webserver

2 participants