Add bounds check on wolfSSL_X509_notBefore and wolfSSL_X509_notAfter #10071

Merged douzzer merged 2 commits into `wolfSSL:master` from `padelsbach:notbefore-notafter-bou…` last week

Conversation Commits 2 Checks Files changed

padelsbach commented 2 weeks ago — Contributor

Description

Fixes zd 21416

Testing

How did you test?

Checklist

- [ ] added tests
- [ ] updated/added doxygen
- [ ] updated appropriate READMEs
- [ ] Updated manual and documentation

padelsbach commented 2 weeks ago • edited — Contributor Author

jenkins retest this please

programsurf commented 2 weeks ago

GitHub PR Comment for #10071

> Post this as a review comment on the PR

Hi, thank you for working on this fix. The bounds check in the getter functions is a good first step and prevents the immediate overflow. However, I found several gaps in the current patch that should be addressed before merging:

1. Root cause not fixed (`internal.c`)

The overflow originates in `CopyDecodedToX509()` where the length is capped at `MAX_DATE_SZ` (32), but the safe maximum for the copy-at-offset-2 pattern in the getter is `CTC_DATE_SIZE - 2` (30):

```
// internal.c:13824 (notBefore) and 13832 (notAfter)
minSz = (int)min(dCert->beforeDate[1], MAX_DATE_SZ);  // allows 31, 32
```

This should be:

```
minSz = (int)min(dCert->beforeDate[1], CTC_DATE_SIZE - 2);  // caps at 30
```

Same issue at lines 14224 and 14234 for attribute certificates.

2. Setter functions not validated (`x509.c:16057-16083`)

`wolfSSL_X509_set_notAfter()` and `wolfSSL_X509_set_notBefore()` accept arbitrary `t->length` without validation:

```
x509->notAfter.length = t->length;    // no bounds check
```

A caller can store `length = 32`, then the getter returns NULL — confusing API behavior. The setter should also validate:

```
if (t->length < 0 || t->length > (int)sizeof(x509->notAfterData) - 2)
    return WOLFSSL_FAILURE;
```

3. NULL return may crash callers

Existing code that calls `wolfSSL_X509_notAfter()` may not expect NULL for a valid `WOLFSSL_X509` with parsed dates. Consider whether a clamp-and-truncate approach would be safer:

```
int cpSz = min(x509->notAfter.length, (int)sizeof(x509->notAfterData) - 2);
XMEMCPY(&x509->notAfterData[2], x509->notAfter.data, cpSz);
x509->notAfterData[1] = (byte)cpSz;
```

This preserves as much date information as possible without overflowing.

Summary

Location	Status
Getter bounds check (this PR)	Fixed
Root cause in CopyDecodedToX509	**Not fixed**
Root cause in attribute cert path	**Not fixed**
Setter validation	**Not fixed**
NULL safety for callers	**Not addressed**

The getter check is defense-in-depth — good to have, but the root cause at the data source should also be fixed to prevent the struct from entering an invalid state.

padelsbach force-pushed the `notbefore-notafter-bounds-check` branch from **`6fb048f`** to **`c51adf7`** 2 weeks ago

Compare

padelsbach commented 2 weeks ago

Contributor Author

Hi **@programsurf**, thanks for the additional comments. These are beyond the scope of the original 2 byte overflow described in the support ticket. However, I've expanded this commit to address most of the additional concerns. I did not address the NULL return (item 3).

padelsbach force-pushed the `notbefore-notafter-bounds-check` branch 3 times, most recently from **`84bb1f0`** to **`d5323a0`** 2 weeks ago

Compare

programsurf commented 2 weeks ago

PR #10071 Review — Second Round (post force-push `d5323a0`)

Thanks for the updated commit — you've addressed items 1, 2, and 4 from my earlier comment. I see you decided not to address item 3 (NULL return safety), which is fair. Here's what I found looking at the new code:

What's fixed

The new `CopyDateToASN1_TIME()` helper is a nice improvement — consolidates the copy logic that was duplicated in 4 places, clamps to buffer size, handles the edge case where `srcDateLen < 2`. Both the regular cert and attribute cert paths now go through it. Getter bounds checks look correct. Test coverage for malicious lengths (255, 128, -1) was added. Overall this is solid work.

Remaining issue: setter/getter bounds mismatch

There's still an off-by-2 inconsistency between the setter and getter that will cause confusing behavior.

The setter checks (x509.c ~line 16075):

```
if (t->length < 0 || t->length > (int)sizeof(x509->notAfter.data)) {
```

`sizeof(x509->notAfter.data)` is `CTC_DATE_SIZE` = 32, so this allows length 0–32.

But the getter checks (x509.c ~line 4449):

```
if (x509->notAfter.length > (int)sizeof(x509->notAfterData) - 2) {
```

`sizeof(notAfterData) - 2` = 30, so this rejects anything above 30.

And `CopyDateToASN1_TIME` also clamps to 30 (`sizeof(data) - 2`).

So if someone calls `wolfSSL_X509_set_notAfter()` with length 31 or 32, the setter happily accepts it, but then `wolfSSL_X509_notAfter()` returns NULL because it can't fit the data into `notAfterData[2:]`. That's a confusing API contract — the set succeeds but the corresponding get fails silently.

The fix is straightforward — the setter should use the same limit:

```
if (t->length < 0 || t->length > (int)sizeof(x509->notAfter.data) - 2) {
```

Same for `set_notBefore`.

Test gap: boundary test doesn't cover the right code path

The boundary test at `CTC_DATE_SIZE`:

```
crafted_time.length = CTC_DATE_SIZE;  // 32
ExpectIntEQ(wolfSSL_X509_set_notAfter(x, &crafted_time), WOLFSSL_SUCCESS);
ExpectNotNull(retrieved = X509_get_notAfter(x));
```

This passes because `X509_get_notAfter()` maps to `wolfSSL_X509_get_notAfter()`, which just returns `&x509->notAfter` directly — no copy, no bounds check. But `wolfSSL_X509_notAfter()` (the one that copies into `notAfterData`) would return NULL for length=32 since `32 > 30`. The test doesn't catch this because it's testing a different getter. Worth adding a `wolfSSL_X509_notAfter()` call here to confirm the behavior at the boundary.

NULL return — acknowledged

You declined to address item 3 (NULL safety for callers), which is fair — it's a design decision. Just noting that existing callers in apps like httpd or haproxy may not expect NULL from `wolfSSL_X509_notAfter()` on a parsed cert, so this could surface as regressions. The setter bounds fix above would reduce the window for this, since the root-cause clamp in `CopyDateToASN1_TIME` already caps at 30.

Minor notes

- `MAX_DATE_SZ` -> `MAX_DATE_SIZE` in `ssl_api_crl_ocsp.c`: both are 32, no functional change — just naming consistency.
- 83 CI checks failing as of the latest force-push. Would be good to triage whether those are related to this change or pre-existing.

TL;DR

1. Setter bounds should be `sizeof(data) - 2` not `sizeof(data)` — matches getter and `CopyDateToASN1_TIME`
2. Boundary test should also exercise `wolfSSL_X509_notAfter()` (not just `X509_get_notAfter`)
3. CI failures need a look

padelsbach force-pushed the `notbefore-notafter-bounds-check` branch 2 times, most recently from **`06e01ca`** to **`2bd4fdb`** 2 weeks ago

Compare

padelsbach commented 2 weeks ago • edited

Contributor Author

jenkins retest this please

padelsbach force-pushed the `notbefore-notafter-bounds-check` branch from **`2bd4fdb`** to **`41b969b`** 2 weeks ago

Compare

padelsbach requested a review from **wolfSSL-Fenrir-bot** last week

padelsbach force-pushed the `notbefore-notafter-bounds-check` branch from **`41b969b`** to **`bcdde5d`** last week

Compare

wolfSSL-Fenrir-bot reviewed last week

View reviewed changes

wolfSSL-Fenrir-bot left a comment

Fenrir Automated Review — PR #10071

Scan targets checked: `src`, `src-bugs`, `src-compliance`

No new issues found in the changed files. ✅

padelsbach force-pushed the `notbefore-notafter-bounds-check` branch from **`bcdde5d`** to **`fc41a6a`** last week

Compare

padelsbach marked this pull request as ready for review last week

padelsbach assigned **wolfSSL-Bot** last week

douzzer commented last week

Contributor

retest this please

padelsbach added 2 commits last week

`Add bounds check on wolfSSL_X509_notBefore and wolfSSL_X509_notAfter` `452652b`

`Add test cases` ✓ `ec9b6cf`

padelsbach force-pushed the `notbefore-notafter-bounds-check` branch from **`fc41a6a`** to **`ec9b6cf`** last week

Compare

douzzer added For This Release Staged labels last week

douzzer approved these changes last week

View reviewed changes

douzzer merged commit **`4dc3470`** into `wolfSSL:master` last week

View details

491 checks passed

Sign up for free **to join this conversation on GitHub**. Already have an account? Sign in to comment

Reviewers

wolfSSL-Fenrir-bot

douzzer ✓

Assignees

wolfSSL-Bot

Labels

For This Release Staged

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

5 participants