Various security fixes and tests #10088

Merged douzzer merged 12 commits into `wolfSSL:master` from `anhu:new_various` 2 days ago

Conversation 30 Commits 12 Checks 478 Files changed 9

anhu commented 2 weeks ago — Member

Fixes:
zd21412
zd21413
zd21414
zd21415
zd21417
zd21418
zd21422
zd21423
zd21426

anhu requested a review from **wolfSSL-Bot** 2 weeks ago

anhu self-assigned this 2 weeks ago

anhu mentioned this pull request 2 weeks ago

Various security fixes and tests #10074

Closed

anhu force-pushed the `new_various` branch from **a0019fb** to **83c456b** 2 weeks ago Compare

Frauschi added the For This Release label 2 weeks ago

anhu assigned **wolfSSL-Bot** and unassigned **anhu** last week

douzzer commented last week — Contributor

retest this please

```
FAIL: scripts/ocsp-stapling2.test
```

douzzer added Staged and removed Staged labels last week

douzzer requested changes last week

View reviewed changes

douzzer left a comment — Contributor

Nit from clang-tidy:

```
[quantum-safe-wolfssl-all-clang-tidy] [33 of 61] [4f295cfb83]
    configure${config_analyzer_note}...   real 0m13.426s  user 0m8.036s  sys 0m6.4
    build...83c456b463 (<anthony@wolfssl.com> 2026-03-27 09:15:34 -0400 34762)
msg, 0xFFFFFFC0u, &res, &key), WC_NO_ERR_TRACE(BAD_FUNC_ARG));
/tmp/tmp.4346_28411/wolfssl_test_workdir.15574/wolfssl/tests/api.c:34762:14: warning:
integer literal has suffix 'u', which is not uppercase [readability-uppercase-
literal-suffix]
34762 |         msg, 0xFFFFFFC0u, &res, &key), WC_NO_ERR_TRACE(BAD_FUNC_ARG));
|              ^         ~
       |                       U
```

douzzer assigned **anhu** last week

anhu commented last week — Member, Author

jenkins retest this please.

douzzer added the Staged label last week

douzzer requested changes last week

View reviewed changes

douzzer left a comment

Contributor

```
Testing configuration:
--enable-ecc --enable-tlsx --enable-supportedcurves --enable-heapmath --enable-
ocspstapling --enable-harden --enable-secure-renegotiation --enable-certgen --enable-
pkcs7 --enable-indef --enable-smallcache --enable-pkcallbacks --enable-cmac --enable-
keygen --enable-poly1305 --enable-chacha --disable-oldtls --disable-sha224 --disable-
errorqueue --enable-aesgcm=small --disable-inline --disable-asm --enable-
opensslextra=x509small --enable-tls13 --enable-des3 --enable-atomicuser --enable-crl
--enable-wpas=small --enable-eccencrypt
Testing DEFAULT: --enable-ecc --enable-tlsx --enable-supportedcurves --enable-
heapmath --enable-ocspstapling --enable-harden --enable-secure-renegotiation --
enable-certgen --enable-pkcs7 --enable-indef --enable-smallcache --enable-pkcallbacks
--enable-cmac --enable-keygen --enable-poly1305 --enable-chacha --disable-oldtls --
disable-sha224 --disable-errorqueue --enable-aesgcm=small --disable-inline --disable-
asm --enable-opensslextra=x509small --enable-tls13 --enable-des3 --enable-atomicuser
--enable-crl --enable-wpas=small --enable-eccencrypt
Configure RESULT = 0
make[2]: warning: -j3 forced in submake: resetting jobserver mode.
In file included from ./wolfssl/wolfcrypt/error-crypt.h:34,
                 from ./wolfssl/error-ssl.h:27,
                 from ./wolfssl/ssl.h:35,
                 from ./tests/unit.h:40,
                 from tests/api.c:32:
tests/api.c: In function ‘test_zd21422_pkcs7_padding_oracle’:
./wolfssl/wolfcrypt/types.h:987:31: error: ‘encodedSz’ may be used uninitialized in
this function [-Werror=maybe-uninitialized]
  987 |     #define XMEMCPY(d,s,l)    memcpy((d),(s),(l))
      |                               ^~~~~~
tests/api.c:34811:9: note: ‘encodedSz’ was declared here
34811 |     int encodedSz;
      |         ^~~~~~~~~
```

```
Testing configuration:
--enable-sp-math-all=small --enable-all --disable-asm --enable-smallstack --enable
32bit CFLAGS="-m32"
The case with CFLAGS + double-quotes
EXTRACTED_CFLAGS = -m32
Testing CONFIG_REMAINDER = --enable-sp-math-all=small --enable-all --disable-asm --
enable-smallstack --enable-32bit
Configure RESULT = 0
```

```
make[2]: warning: -j3 forced in submake: resetting jobserver mode.
In file included from ./wolfssl/wolfcrypt/error-crypt.h:34,
                 from ./wolfssl/error-ssl.h:27,
                 from ./wolfssl/ssl.h:35,
                 from ./tests/unit.h:40,
                 from tests/api.c:32:
tests/api.c: In function ‘test_zd21422_pkcs7_padding_oracle’:
./wolfssl/wolfcrypt/types.h:987:31: error: ‘encodedSz’ may be used uninitialized in
this function [-Werror=maybe-uninitialized]
  987 |     #define XMEMCPY(d,s,l)    memcpy((d),(s),(l))
      |                               ^~~~~~
tests/api.c:34811:9: note: ‘encodedSz’ was declared here
34811 |     int encodedSz;
      |         ^~~~~~~~~
```

```
Testing CONFIG_REMAINDER = CC=clang   --enable-opensslextra --enable-des3 --enable
--enable-ecc --enable-dtls --enable-aesgcm --enable-aesccm --enable-sniffer --enabl
psk --enable-camellia --enable-sha512 --enable-crl --enable-ocsp --enable-savesession
--enable-savecert --enable-atomicuser --enable-pkcallbacks --enable-scep
32
Configure RESULT = 0
33
34
make[2]: warning: -j3 forced in submake: resetting jobserver mode.
35
In file included from src/ssl.c:288:
36
./src/ssl_sess.c:530:34: error: unused variable 's' [-Werror,-Wunused-variable]
37
                WOLFSSL_SESSION* s = &SessionCache[i].Sessions[j];
38
                                 ^
39
./src/ssl_sess.c:705:34: error: unused variable 's' [-Werror,-Wunused-variable]
40
                WOLFSSL_SESSION* s = &SessionCache[i].Sessions[j];
41
                                 ^
```

and others

douzzer removed the Staged label last week

anhu force-pushed the `new_various` branch from **179c337** to **a33731a** last week Compare

JacobBarthelmeh reviewed last week

View reviewed changes

`wolfcrypt/src/ecc.c` Outdated — Show resolved

anhu requested review from **JacobBarthelmeh** and **douzzer** last week

anhu removed their assignment last week

dgarske requested changes 3 days ago

View reviewed changes

`src/ssl_sess.c` Outdated — Show resolved

`wolfcrypt/src/evp.c` Outdated — Show resolved

`wolfcrypt/src/pkcs7.c` Outdated — Show resolved

`wolfcrypt/src/pkcs7.c` Outdated — Show resolved

dgarske assigned **anhu** 3 days ago

anhu requested a review from **dgarske** 3 days ago

anhu assigned **wolfSSL-Bot** and unassigned **anhu** and **wolfSSL-Bot** 3 days ago

douzzer previously requested changes 3 days ago

View reviewed changes

douzzer left a comment — Contributor

AI review -- several of these should be fixed before merge and are easy:

PR Summary

This is a security-fix batch: 9 ZD-tracked vulnerabilities (21412–21426), each with a targeted fix, plus 5 regression tests. The fixes span ML-DSA, TLS 1.3 PQC key exchange, PKCS7/CMS, ECC import, session cache serialization, and EVP PKEY printing.

Findings

1. PKCS7 padding check is not actually constant-time (ZD 21422 — both paths)

The comment says "Constant-time check all padding bytes," but the loop iterates exactly `padLen` times:

```
for (padIndex = encryptedContentSz - padLen;
     padIndex < encryptedContentSz; padIndex++) {
    padCheck |= encryptedContent[padIndex] ^ padLen;
}
```

The iteration count is data-dependent — it leaks `padLen` (i.e. the last plaintext byte) through timing. A true constant-time check would iterate over the full final block and use a mask to conditionally accumulate. The early-bail checks (`padLen == 0 || padLen > expBlockSz`) also leak through control flow, though those reject structurally invalid values.

Practical impact is low for PKCS7/CMS, which is typically not a network-facing oracle the way TLS CBC is (Vaudenay/Lucky13). The fix is a substantial improvement over the pre-existing single-last-byte check. But the comment should either be corrected or the implementation made truly constant-time, depending on threat model.

2. Session cache restore: `ticketLen` not sanitized (ZD 21423)

After the raw `XMEMCPY` / `XFREAD` , the fix resets `ticket = staticTicket` and `ticketLenAlloc = 0` , but does **not** cap `ticketLen` . If the serialized session had a dynamically-allocated ticket with `ticketLen > SESSION_TICKET_LEN` , the restored session now has `ticket` pointing to `staticTicket` (size `SESSION_TICKET_LEN`) with a `ticketLen` that exceeds it. Any subsequent read of `ticket[0..ticketLen-1]` would be OOB from `staticTicket` .

This is arguably pre-existing in the save/restore mechanism (it serializes raw structs including stale pointer values), but this fix makes the inconsistency more dangerous: the pointer is now valid (so no obvious crash), but the length can exceed the buffer (silent OOB). Consider adding:

```
#ifdef HAVE_SESSION_TICKET
    s->ticket = s->staticTicket;
    s->ticketLenAlloc = 0;
    if (s->ticketLen > SESSION_TICKET_LEN)
        s->ticketLen = SESSION_TICKET_LEN;
#endif
```

3. Incomplete `wc_*_delete()` coverage (ZD 21415)

The heap-before-free pattern is fixed for `curve25519`, `ed25519`, and `dilithium`. If `ed448`, `curve448`, or any other key types have the same `wc_*_delete()` → `wc_*_free()` → `ForceZero` → `XFREE(key, key->heap, ...)` pattern, they remain vulnerable. Not a regression from this PR, but worth an audit pass.

4. `MP_DECIMAL_MAX_LEN` naming is misleading (ZD 21426)

The define is 24, which is fine for RSA/DSA/DH/EC exponents and small parameters. But the name `MP_DECIMAL_MAX_LEN` reads as "maximum decimal representation length for any MP integer," which it clearly isn't (a 2048-bit MP integer needs ~617 decimal digits). If any future code uses this define for general mp_int-to-decimal conversion, it will overflow. A name like `EVP_EXPONENT_PRINT_MAX` or similar would be less error-prone.

5. Test callback `oriDecryptCb_zd21414` has an unchecked OOB read (test-only)

```
keySz = oriValue[1];
if (*decryptedKeySz < keySz) return -1;
XMEMCPY(decryptedKey, oriValue + 2, keySz);
```

There's a check `if (oriValueSz < 2) return -1;` but no check that `keySz <= oriValueSz - 2`. If the ORI value data is shorter than `oriValue[1]` claims, this reads OOB from `oriValue`. This is test-only code and the crafted DER is designed to fail at the OID bounds check before the callback is reached, so it's not a production issue — but a fuzzer hitting this callback with a different input could cause a confusing test crash. A one-liner `if (keySz > oriValueSz - 2) return -1;` would harden it.

6. No regression test for ZD 21413 (TLS 1.3 PQC key share over-read)

This is the most critical fix — a malicious TLS 1.3 server can trigger it remotely. The others all require local/crafted input. Understandable that it's hard to unit-test without TLS handshake infrastructure, but worth noting as a gap.

Fixes that look clean

- **ZD 21412** (dilithium hashLen check) — straightforward, mirrors sign-path check.
- **ZD 21413** (TLS PQC `keLen < ctSz` check) — correct, positioned before the `Decapsulate` call, proper error code.
- **ZD 21414** (ORI OID bounds check) — correct, `oriOIDSz > MAX_OID_SZ` before `XMEMCPY` into `oriOID[MAX_OID_SZ]`.
- **ZD 21415** (heap-before-free) — correct pattern, clean across all three files.
- **ZD 21417** (integer overflow check) — both the `data2Len > (UINT32_MAX - data1Len)` and the `msgLen > UINT32_MAX / 2` guards are correct.
- **ZD 21418** (ECC point validation for untrusted imports) — clean. In the `#else` of `WOLFSSL_VALIDATE_ECC_IMPORT`, gates on `untrusted` so trusted imports (cert parsing) are unaffected.

Binary blob check: The `malformed[]` array in the ZD 21414 test is a verifiable crafted CMS EnvelopedData with visible DER structure — OIDs, the patched 0x40 length byte, 54 ASCII 'X' (0x58) pad bytes, AES-256-CBC OID, IV, and ciphertext. No opaque or suspicious blobs.

Verdict

No security regressions introduced. The fixes are well-targeted and the tests are solid. Findings #1 (constant-time comment) and #2 (`ticketLen` cap) are the most actionable items before merge.

douzzer assigned **anhu** 3 days ago

anhu commented 3 days ago — Member — Author

448 variants don't have XFREE() in them.

17 hidden items

Load more…

anhu and others added 2 commits 2 days ago

`Add regression tests for fixes` `2e32094`

`Remove UTF-8 chars` ··· ✕ `c335f7d`

JacobBarthelmeh force-pushed the `new_various` branch from **`7a9557c`** to **`c335f7d`** 2 days ago Compare

JacobBarthelmeh assigned **JacobBarthelmeh** and unassigned **wolfSSL-Bot** 2 days ago

`adjust test case return value check after rebase` ✕ `ad1cc4e`

JacobBarthelmeh commented 2 days ago — Contributor

After the rebase these commits where removed:

- `feb8782` fixed in (**Improved ECC curve validation** #10133)
- `8a60289` fixed in (**Additional fixes** #10116)
- `978c37f` fixed in (**20260403-WC_FIPS_186** #10131)

dgarske requested changes 2 days ago

View reviewed changes

`src/ssl_sess.c` Outdated — Show resolved

dgarske assigned **anhu** and **JacobBarthelmeh** and unassigned **JacobBarthelmeh** and **anhu** 2 days ago

`refactor sanity pointer set of session and clean up macro guards` ✕ `ecfd117`

JacobBarthelmeh requested a review from **Copilot** 2 days ago

dgarske previously approved these changes 2 days ago

View reviewed changes

dgarske dismissed **douzzer**'s stale review 2 days ago

Fixed

Copilot started reviewing on behalf of **JacobBarthelmeh** 2 days ago

View session

JacobBarthelmeh requested a review from **wolfSSL-Fenrir-bot** 2 days ago

Copilot AI reviewed 2 days ago

View reviewed changes

Copilot AI left a comment

Contributor

Pull request overview

This PR delivers a set of security hardening changes across wolfCrypt and TLS/session handling, and adds regression tests covering the newly enforced validation behaviors (referenced by the listed zd* items).

Changes:

- Harden PKCS#7 CBC padding validation by rejecting invalid padding sizes and validating all padding bytes during decode.
- Add additional bounds/overflow validation in PQC key share decapsulation and Dilithium hashing/verification paths.
- Improve safety in key object delete helpers (preserve heap before freeing) and add new API-level regression tests.

Reviewed changes

Copilot reviewed 9 out of 9 changed files in this pull request and generated 5 comments.

▶ Show a summary per file

💡 Add Copilot custom instructions for smarter, more guided reviews. Learn how to get started.

`wolfssl/wolfcrypt/types.h` Outdated

Show resolved

`wolfcrypt/src/pkcs7.c` Outdated — Show resolved

`wolfcrypt/src/pkcs7.c` Outdated — Show resolved

`tests/api.c` — Show resolved

`src/ssl_sess.c` — Show resolved

JacobBarthelmeh commented 2 days ago — Contributor

Retest this please Jenkins : Windows ACVP test failure

wolfSSL-Fenrir-bot reviewed 2 days ago

View reviewed changes

wolfSSL-Fenrir-bot left a comment

Fenrir Automated Review — PR #10088

Scan targets checked: `wolfcrypt-api_misuse`, `wolfcrypt-bugs`, `wolfcrypt-compliance`, `wolfcrypt-concurrency`, `wolfcrypt-consttime`, `wolfcrypt-defaults`, `wolfcrypt-mutation`, `wolfcrypt-portability`, `wolfcrypt-proptest`, `wolfcrypt-src`, `wolfcrypt-zeroize`, `wolfssl-bugs`, `wolfssl-compliance`, `wolfssl-consttime`, `wolfssl-defaults`, `wolfssl-mutation`, `wolfssl-proptest`, `wolfssl-src`, `wolfssl-zeroize`
Findings: 4
2 finding(s) posted as inline comments (see file-level comments below)

Medium (2)

PKCS7 EnvelopedData padding check is not constant-time

File: `src/pkcs7.c:13264-13282`
Function: `wc_PKCS7_DecodeEnvelopedData`
Category: Secret-dependent branches

The new PKCS#7 padding validation in `wc_PKCS7_DecodeEnvelopedData` introduces two timing side channels on secret data:

1. **Early-exit branch on secret value**: The check `if (padLen == 0 || padLen > expBlockSz || padLen > encryptedContentSz)` branches directly on `padLen`, which is the last byte of decrypted ciphertext — secret data. This creates a distinguishable fast path (no loop executed) vs slow path (loop executed).

2. **Variable-iteration loop**: The padding verification loop `for (padIndex = encryptedContentSz - padLen; padIndex < encryptedContentSz; padIndex++)` iterates exactly `padLen` times, where `padLen` ranges from 1 to `expBlockSz` (e.g., 1-16 for AES). An attacker measuring timing can distinguish between different padding byte values.

The code comment acknowledges this: "Better implementation would be to run through the entire block." Since this fix specifically targets a padding oracle vulnerability (ZD 21422), the remaining timing side channel undermines the fix. A constant-time implementation should iterate over the full block and use bitwise masking to validate padding bytes regardless of `padLen`, avoiding early exits.

```
padLen = encryptedContent[encryptedContentSz-1];

if (padLen == 0 || padLen > expBlockSz ||
    padLen > encryptedContentSz) {
    ret = BUFFER_E;
    break;
}

/* Check all padding bytes. Better implementation would be to run
 * through the entire block. */
for (padIndex = encryptedContentSz - padLen;
     padIndex < encryptedContentSz; padIndex++) {
    padCheck |= encryptedContent[padIndex] ^ padLen;
}
if (padCheck != 0) {
    ret = BUFFER_E;
    break;
}
```

Recommendation: Replace the early-exit check and variable-length loop with a constant-time full-block scan. For example, iterate over all `expBlockSz` bytes of the last block, using bitwise operations to mask which bytes should match `padLen`. Use `ctMaskGTE` / `ctMaskLTE` (or equivalent wolfSSL constant-time helpers from `wolfcrypt/src/misc.c`) to build the expected padding mask without branching on `padLen`. Defer the `padLen == 0 || padLen > expBlockSz` check into the constant-time logic by OR-ing the error into `padCheck` rather than breaking early.

PKCS7 EncryptedData padding check is not constant-time

File: `src/pkcs7.c:15331-15357`
Function: `wc_PKCS7_DecodeEncryptedData`
Category: Secret-dependent branches

The same non-constant-time padding validation pattern appears in `wc_PKCS7_DecodeEncryptedData`. The check `if (padLen == 0 || padLen > expBlockSz || padLen > encryptedContentSz)` branches on the secret-derived `padLen`, and the subsequent loop iterates `padLen` times rather than a fixed number of iterations. This creates the same timing side channel as the `wc_PKCS7_DecodeEnvelopedData` instance — an attacker can distinguish different `padLen` values via timing measurement, potentially enabling a padding oracle attack despite the fix.

```
padLen = encryptedContent[encryptedContentSz-1];

if (padLen == 0 || padLen > expBlockSz ||
    padLen > encryptedContentSz) {
    WOLFSSL_MSG("Bad padding size found");
    ret = BUFFER_E;
    XFREE(encryptedContent, pkcs7->heap, DYNAMIC_TYPE_PKCS7);
    break;
}

/* Check all padding bytes. Better implementation would be to
 * run through the entire block. */
for (padIndex = encryptedContentSz - padLen;
     padIndex < encryptedContentSz; padIndex++) {
     padCheck |= encryptedContent[padIndex] ^ padLen;
}
if (padCheck != 0) {
    WOLFSSL_MSG("Bad padding bytes found");
    ret = BUFFER_E;
    XFREE(encryptedContent, pkcs7->heap, DYNAMIC_TYPE_PKCS7);
    break;
}
```

Recommendation: Apply the same constant-time fix as recommended for `wc_PKCS7_DecodeEnvelopedData`: iterate over all `expBlockSz` bytes using bitwise masking to validate padding without leaking `padLen` through timing. Both functions should share a common constant-time padding validation helper.

This review was generated automatically by Fenrir. Findings are non-blocking.

wolfcrypt/src/pkcs7.c — Show resolved

src/ssl_sess.c — Show resolved

make the padding check constant time and move evp exponent print size… d1c6423

…

JacobBarthelmeh dismissed **dgarske**'s stale review via `d1c6423` 2 days ago

dgarske previously approved these changes 2 days ago

View reviewed changes

dgarske assigned **wolfSSL-Bot** 2 days ago

JacobBarthelmeh commented 2 days ago — Contributor

Retest this please Jenkins : Windows ACVP fail

`add adjustment for review` `4fd0df4`

JacobBarthelmeh dismissed **dgarske**'s stale review via `4fd0df4` 2 days ago

douzzer approved these changes 2 days ago

View reviewed changes

douzzer merged commit **`750f3b1`** into `wolfSSL:master` 2 days ago

View details

470 of 477 checks passed

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

Reviewers

- **Frauschi**
- **Copilot**
- **wolfSSL-Fenrir-bot**
- **douzzer**
- **dgarske**

wolfSSL-Bot

JacobBarthelmeh

Assignees

JacobBarthelmeh

wolfSSL-Bot

Labels

For This Release

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

9 participants