Memory Leak in crypto_handle_incrementing_nontransmitted_counter Function of CryptoLib

Moderate **jlucas9** published **GHSA-p38w-p2r8-g6g5** 5 days ago

Package	Affected versions	Patched versions
No package listed	<= 1.3.3	None

Description

Summary

A memory leak vulnerability was identified in the `crypto_handle_incrementing_nontransmitted_counter` function of CryptoLib. This vulnerability can lead to resource exhaustion and degraded system performance over time, particularly in long-running processes or systems processing large volumes of data.

Details

The vulnerability is present in the `crypto_handle_incrementing_nontransmitted_counter` function within `crypto_tc.c`. The function allocates memory using `malloc` without ensuring the allocated memory is always freed:

```
uint8_t *temp_counter = malloc(src_full_len);
memcpy(temp_counter, src, src_full_len);

// Increment temp_counter Until Transmitted Portion Matches Frame.
uint8_t counter_matches = CRYPTO_TRUE;
for (int i = 0; i < window; i++)
{
    Crypto_increment(temp_counter, src_full_len);
    for (int x = (src_full_len - transmitted_len); x < src_full_len;
    {
        if (temp_counter[x] != dest[x])
        {
            counter_matches = CRYPTO_FALSE;
            break;
        }
```

Severity

Moderate

CVE ID

CVE-2025-29910

Weaknesses

CWE-401

Credits

 **mirkobitetto** Finder

 Coordinator **danmaam**

 Coordinator **juriSacchetta**

```
    }
    if (counter_matches == CRYPTO_TRUE)
    {
        break;
    }
}

// Incorrect free logic
if (!temp_counter)
    free(temp_counter);
```

The condition `if (!temp_counter)` is incorrect. Since `temp_counter` is not `NULL`, the `free` statement is never executed, causing a memory leak. This issue was detected by AddressSanitizer (ASan) with the following output:

```
==1156133==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 2 byte(s) in 1 object(s) allocated from:
    #0 0x62a980e93859 in malloc
    (/home/mirko/Documents/tesi/CryptoLib/build-
asan/test/fuzz_harness+0x162859)
    #1 0x7c15825d3099 in
crypto_handle_incrementing_nontransmitted_counter

/home/mirko/Documents/tesi/CryptoLib/src/core/crypto_tc.c:2145:29
    #2 0x7c15825ce2c9 in Crypto_TC_Nontransmitted_SN_Increment

/home/mirko/Documents/tesi/CryptoLib/src/core/crypto_tc.c:1301:13
    #3 0x7c15825c9cba in Crypto_TC_ProcessSecurity_Cam

/home/mirko/Documents/tesi/CryptoLib/src/core/crypto_tc.c:1911:14
    #4 0x7c15825c5d0d in Crypto_TC_ProcessSecurity

/home/mirko/Documents/tesi/CryptoLib/src/core/crypto_tc.c:1216:12

SUMMARY: AddressSanitizer: 2 byte(s) leaked in 1 allocation(s).
```

PoC

1. Compile CryptoLib with AddressSanitizer enabled (`-fsanitize=address`).
2. Pass the following crafted input to the `Crypto_TC_ProcessSecurity` function, which will eventually call `crypto_handle_incrementing_nontransmitted_counter` :

```
08 03 00 02 00 0B 00 0A FD 02 02 00 08 03 00 00 54 00 00 13
```

3. Observe ASan logs, showing a memory leak of 2 bytes due to the unfreed `temp_counter` variable.

Impact

- **Memory Leak (CWE-401)**: This issue can lead to resource exhaustion, reduced system performance, and potentially a Denial of Service (DoS) in environments where CryptoLib is used in long-running processes or with large volumes of data.
- **Affected Systems**: Any system using CryptoLib, especially those handling high-throughput or continuous data streams, could be impacted.