EO/EE 5.7 - Security Vulnerabilities concern on Couchbase

Nuance Development team analyzed the customer's concern related to Security Vulnerabilities on Couchbase and provided the following feedback:

In short, below are some facts which together prove that Nuance already use a Couchbase version which has the fix for the given CVE: CVE-2013-7239

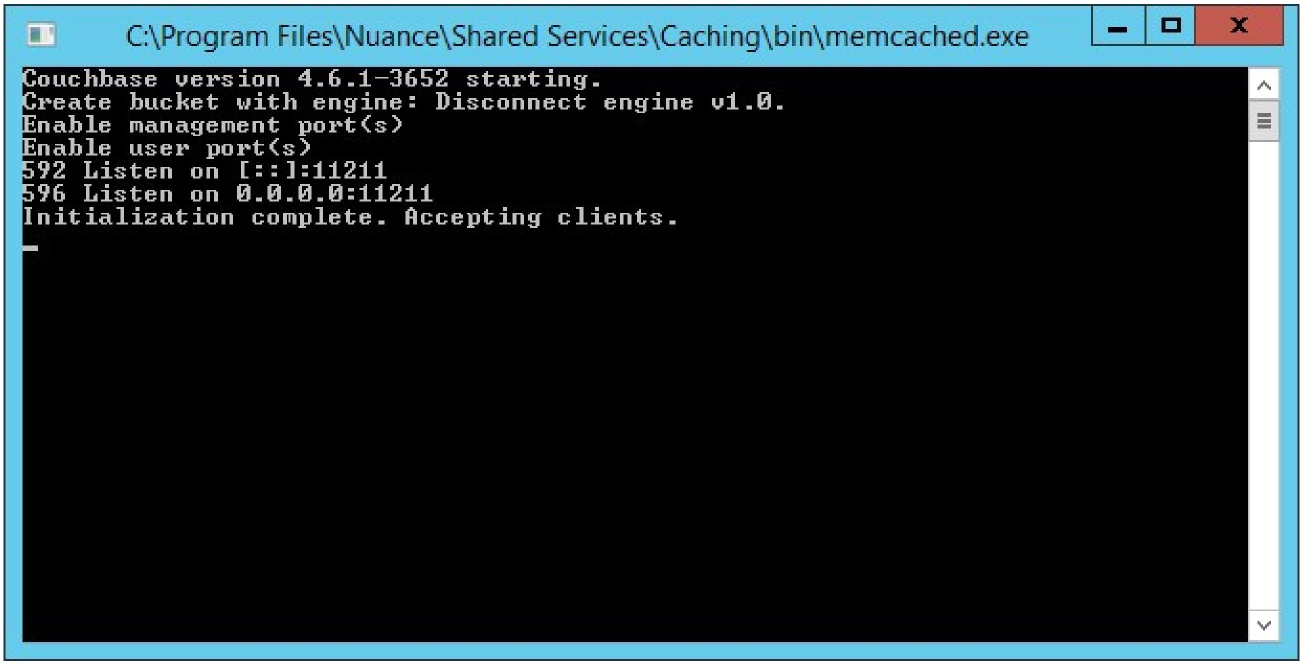
1. Equitrac uses Couchbase v4.6.1-3652, which was released in March 2017.
2. The Memcached issue Customer reported was fixed in Memcached codebase in December 2013, with Memcached v1.4.17.
3. Couchbase uses its own fork of Memcached, which differs from the original Memcached code base.
4. The original Memcached fix can be found in the Couchbase version.
5. This fix is in there since at least September 29, 2015.

In Summary, the original problem was fixed in 2013, and Nuance can prove that the fix is in the Couchbase repo since at least 2015, and Equitrac uses a build of Couchbase from 2017.

Here are the details:

Equitrac uses Couchbase v4.6.1-3652, which was released in March 2017.

Running the memcached.exe file of the Equitrac installation shows this window:



C:\Program Files\Nuance\Shared Services\Caching\bin\memcached.exe

```
Couchbase version 4.6.1-3652 starting.
Create bucket with engine: Disconnect engine v1.0.
Enable management port(s)
Enable user port(s)
592 Listen on [::]:11211
596 Listen on 0.0.0.0:11211
Initialization complete. Accepting clients.
_
```

Please note that, according to the official Couchbase homepage, 4.6.1 was released in March 2017.
See: https://developer.couchbase.com/documentation/server/4.6/release-notes/relnotes.html

"Couchbase Server 4.6.1, released in March 2017, is the first maintenance release in the 4.6.x series for Couchbase Server. This release has fixes related to N1QL query, XDCR, indexing, and backup."

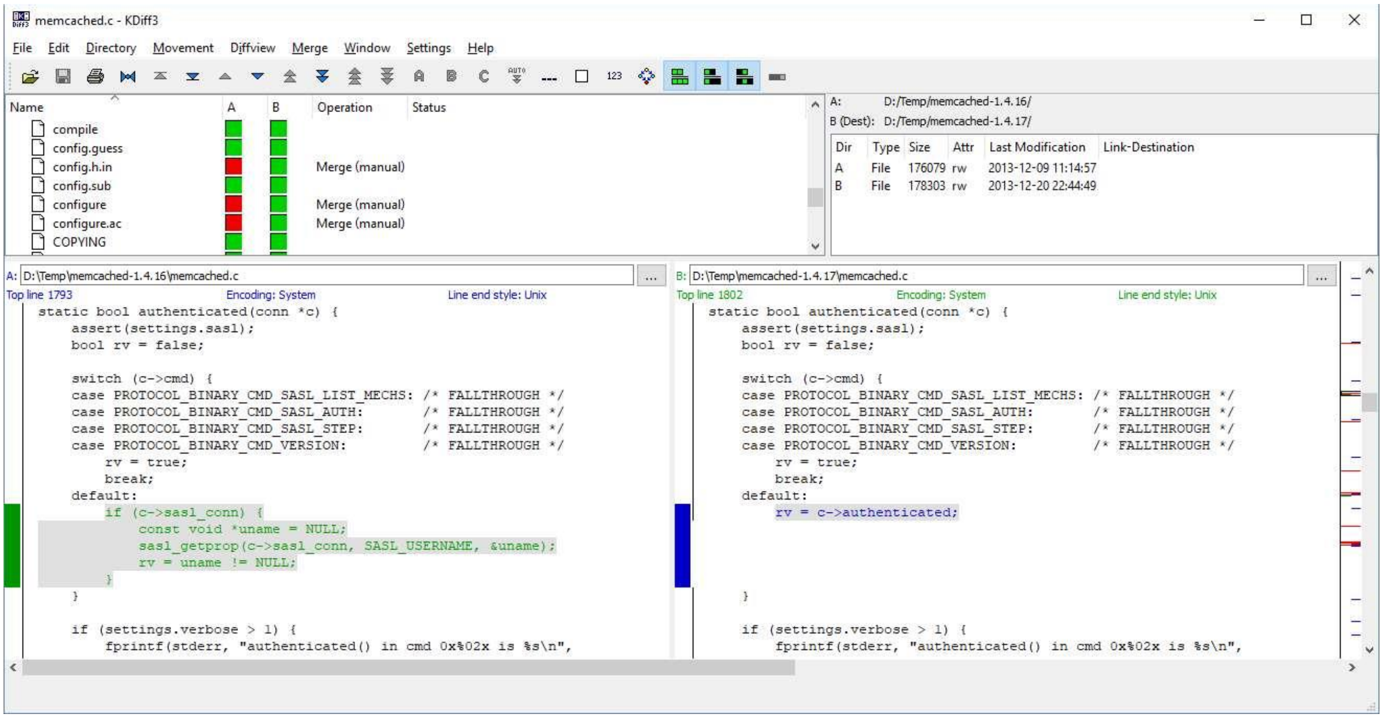
The Memcached issue Customer reported was fixed in Memcached codebase in December 2013, with Memcached v1.4.17.

According to the official Memcached bug entry, the problem is that Memcached can be accessed after a failed authentication.
See: https://code.google.com/archive/p/memcached/issues/316

Also, this page describes that it was fixed with Memcached v1.4.17. The release notes for v1.4.17 really describes it.
See: https://github.com/memcached/memcached/wiki/ReleaseNotes1417

"The other notable bug is a SASL authentication bypass glitch. If a client makes an invalid request with SASL credentials, it will initially fail.
However if you issue a second request with bad SASL credentials, it will authenticate. This has now been fixed."

Comparing the Memcached source codes of v1.4.16 and v1.4.17 does not have too many changes, and we can spot the most important one (related to this bug) in Memcached c:



Name	A	B	Operation	Status
compile	🟩	🟩		
config.guess	🟩	🟩		
config.h.in	🟥	🟩	Merge (manual)	
config.sub	🟩	🟩		
configure	🟥	🟩	Merge (manual)	
configure.ac	🟥	🟩	Merge (manual)	
COPYING	🟩	🟩		

A: D:/Temp/memcached-1.4.16/
B (Dest): D:/Temp/memcached-1.4.17/

Dir	Type	Size	Attr	Last Modification	Link-Destination
A	File	176079	rw	2013-12-09 11:14:57	
B	File	178303	rw	2013-12-20 22:44:49	

A: D:\Temp\memcached-1.4.16\memcached.c
Top line 1793 Encoding: System Line end style: Unix

```
static bool authenticated(conn *c) {
    assert(settings.sasl);
    bool rv = false;

    switch (c->cmd) {
    case PROTOCOL_BINARY_CMD_SASL_LIST_MECHS: /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_SASL_AUTH:       /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_SASL_STEP:       /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_VERSION:         /* FALLTHROUGH */
        rv = true;
        break;
    default:
        if (c->sasl_conn) {
            const void *uname = NULL;
            sasl_getprop(c->sasl_conn, SASL_USERNAME, &uname);
            rv = uname != NULL;
        }
    }

    if (settings.verbose > 1) {
        fprintf(stderr, "authenticated() in cmd 0x%02x is %s\n",
```

B: D:\Temp\memcached-1.4.17\memcached.c
Top line 1802 Encoding: System Line end style: Unix

```
static bool authenticated(conn *c) {
    assert(settings.sasl);
    bool rv = false;

    switch (c->cmd) {
    case PROTOCOL_BINARY_CMD_SASL_LIST_MECHS: /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_SASL_AUTH:       /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_SASL_STEP:       /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_VERSION:         /* FALLTHROUGH */
        rv = true;
        break;
    default:
        rv = c->authenticated;




    }

    if (settings.verbose > 1) {
        fprintf(stderr, "authenticated() in cmd 0x%02x is %s\n",
```

Couchbase uses its own fork of Memcached, which differs from the original Memcached code base.

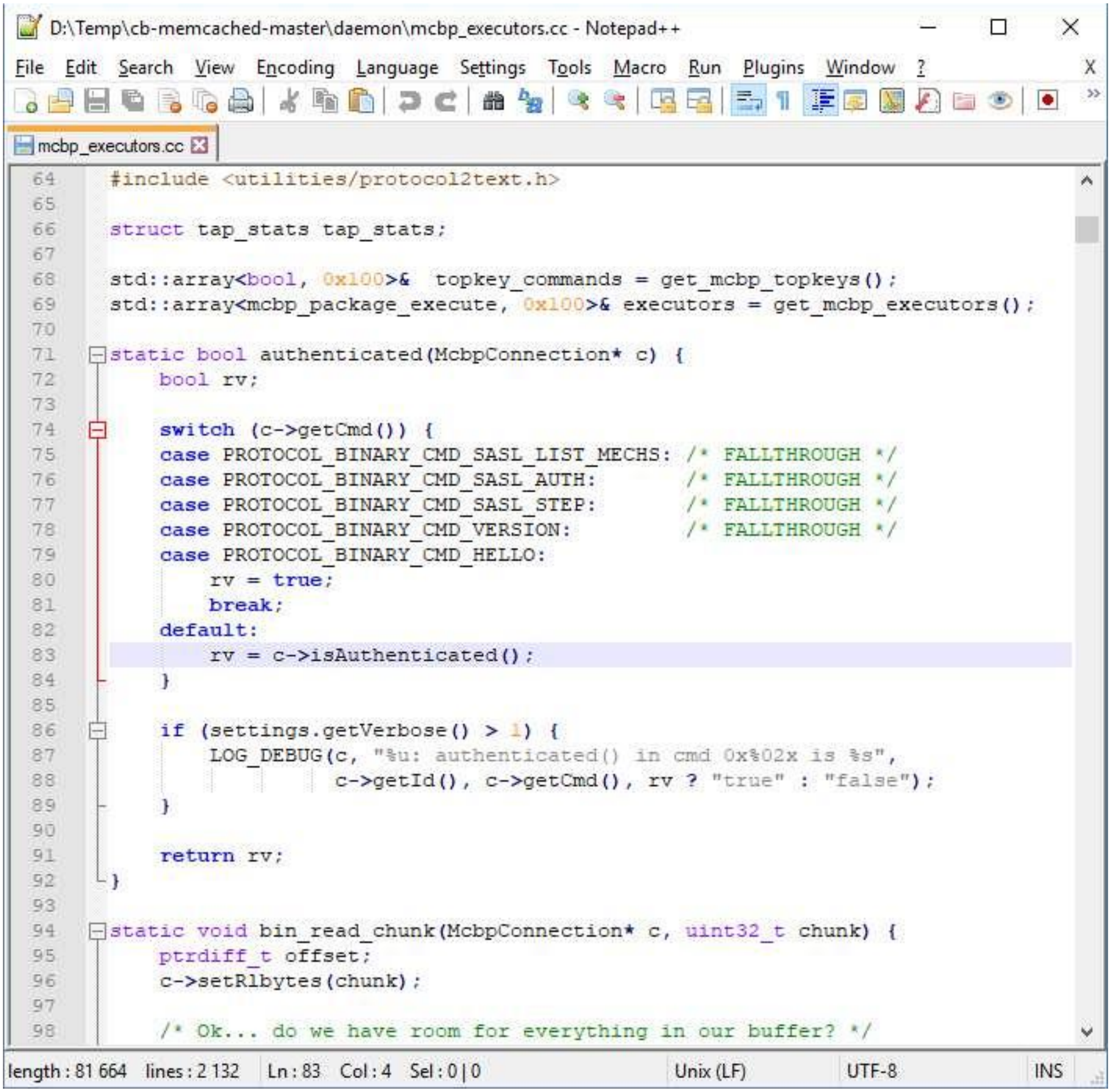
According to the Couchbase source code repository, their fork of Memcached is heavily rewritten.
See: https://github.com/couchbase/memcached

"Welcome to the Couchbase Memcached project.
This started as Couchbase's fork of the upstream Memcached project. It has subsequently evolved since then, so while it shares a name with the upstream project many other things have changed, apart from the name :) For now it's simpler to consider this as the frontend of the Couchbase key-value engine.
The primary backend of KV-engine is the eventually persistent engine - ep-engine."

Comparing the file set, we can see that the original Memcached project has almost everything in one big Memcached.c file, while the Couchbase version refactored this into multiple smaller files.

The original Memcached fix can be found in the Couchbase version.

Even though the fileset is different, we can find the Memcached methods in the Couchbase fileset. For example, the static bool authenticated(conn *c) method can be easily found in the mcbp_executors.cc file of the Couchbase version as static bool authenticated(McbpConnection* c), and we can see that it is the fixed version:



```cpp
#include <utilities/protocol2text.h>

struct tap_stats tap_stats;

std::array<bool, 0x100>&  topkey_commands = get_mcbp_topkeys();
std::array<mcbp_package_execute, 0x100>& executors = get_mcbp_executors();

static bool authenticated(McbpConnection* c) {
    bool rv;

    switch (c->getCmd()) {
    case PROTOCOL_BINARY_CMD_SASL_LIST_MECHS: /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_SASL_AUTH:       /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_SASL_STEP:       /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_VERSION:         /* FALLTHROUGH */
    case PROTOCOL_BINARY_CMD_HELLO:
        rv = true;
        break;
    default:
        rv = c->isAuthenticated();
    }

    if (settings.getVerbose() > 1) {
        LOG_DEBUG(c, "%u: authenticated() in cmd 0x%02x is %s",
                  c->getId(), c->getCmd(), rv ? "true" : "false");
    }

    return rv;
}

static void bin_read_chunk(McbpConnection* c, uint32_t chunk) {
    ptrdiff_t offset;
    c->setRlbytes(chunk);

    /* Ok... do we have room for everything in our buffer? */
```

This fix is in there since at least September 29, 2015.

The Couchbase source code repository shows that this fix is in there since ages. We could track it back up till September 2015, when this split file was created, and it already contained the fix.
See: https://github.com/couchbase/memcached/commit/8917d2cbc1abf1648e864836b279c6f262b80020#diff-75eab956de2cfcf144585628e54d835c

Conclusion:

Putting all these together, Nuance can safely say that the Couchbase version we use in Equitrac already contains the fix for the mentioned CVE.