POC1 analysis:

Breakthrough Type: Cross-Border Visit

First allocate maxCells to 5 size memory through bitmap_cache_new

```
rdpBitmapCache* bitmap_cache_new(rdpSettings* settings)
{
    int i;
    rdpBitmapCache* bitmapCache;
    bitmapCache = (rdpBitmapCache*) calloc(1, sizeof(rdpBitmapCache));

    if (!bitmapCache)
        return NULL;

    bitmapCache->settings = settings;
    bitmapCache->update = ((freerdp*) settings->instance)->update;
    bitmapCache->context = bitmapCache->update->context;
    bitmapCache->maxCells = settings->BitmapCacheV2NumCells;
    bitmapCache->cells = (BITMAP_V2_CELL*) calloc(bitmapCache->maxCells,
                         sizeof(BITMAP_V2_CELL));
```

When accessing bitmapCache-> cells via bitmap_cache_get function

Pass judgment

if (id> bitmapCache-> maxCells)

{

WLog_ERR (TAG, "get invalid bitmap cell id:%" PRIu32 "", id);

return NULL;

}

The maximum limit of 5 is actually greater than the maximum of 5, causing the array to exceed the boundary.

```
rdpBitmap* bitmap_cache_get(rdpBitmapCache* bitmapCache, UINT32 id,
                            UINT32 index)
{
    rdpBitmap* bitmap;

    if (id > bitmapCache->maxCells)
    {
        WLog_ERR(TAG,  "get invalid bitmap cell id: %"PRIu32"", id);
        return NULL;
    }

    if (index == BITMAP_CACHE_WAITING_LIST_INDEX)
    {
        index = bitmapCache->cells[id].number;
    }
    else if (index > bitmapCache->cells[id].number)
    {
        WLog_ERR(TAG,  "get invalid bitmap index %"PRIu32" in cell id: %"PRIu32"", index, id);
        return NULL;
    }
```