## Integer overflow 1:

There is an integer overflow at Freerdp Surface

```
static UINT gdi_CreateSurface(RdpgfxClientContext* context,
                              const RDPGFX_CREATE_SURFACE_PDU* createSurface)
   UINT rc = ERROR_INTERNAL_ERROR;
   rdpGdi* gdi = (rdpGdi*)context->custom;
   EnterCriticalSection(&context->mux);
   surface = (gdiGfxSurface*)calloc(1, sizeof(gdiGfxSurface));
   if (!surface)
        goto fail;
    surface->codecs = gdi->context->codecs;
    if (!surface->codecs)
        free(surface);
   surface->surfaceId = createSurface->surfaceId;
   surface->width = gfx align scanline(createSurface->width, 16);
   surface->height = gfx_align_scanline(createSurface->height, 16);
    surface->mappedWidth = createSurface->width;
    surface->mappedHeight = createSurface->height;
    surface->outputTargetWidth = createSurface->width;
    surface->outputTargetHeight = createSurface->height;
```

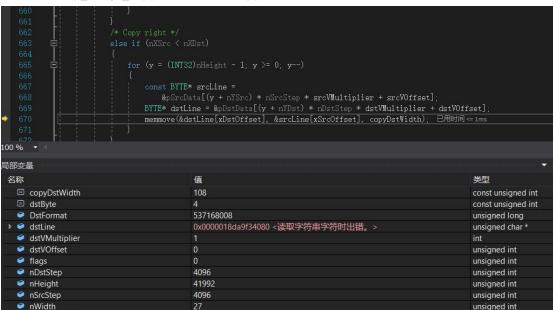
Gdi\_createsurface will allocate a memory of height size

```
surface->scanline = gfx_align_scanline(surface->width * 4, 16);
surface->data = (BYTE*)_aligned_malloc(surface->scanline * surface->height, 16);
```

When parsing surface data, the size of height will be reset and its length is not checked

## Also gdi\_CacheToSurface

## When freerdp\_image\_copy is called, the copy will cause overflow



## Integer overflow 2:

if  $rect \rightarrow bottom < rect \rightarrow top$ 

```
| Triangle | Triangle
```

```
⊟static UINT gdi_SolidFil1(RdpgfxClientContext* context, const RDPGFX_SOLID_FILL_PDU* solidFill)
              UINT status = ERROR_INTERNAL_ERROR;
              UINT16 index:
              UINT32 nWidth, nHeight;
              EnterCriticalSection(&context->mux)
              surface = (gdiGfxSurface*)context->GetSurfaceData(context, solidFill->surfaceId);
              b = solidFill->fillPixel.B;
              r = solidFill->fillPixel.R;
              color = FreeRDPGetColor(surface->format, r, g, b, a);
              for (index = 0; index < solidFill->fillRectCount; index++)
              const UINT32 bpp = GetBytesPerPixel(DstFormat);
              BYTE* pFirstDstLine = &pDstData[nYDst * nDstStep];
             BYTE* pFirstDstLineXOffset = &pFirstDstLine[nXDst * bpp];
                   BYTE* pDst = &pFirstDstLine[(x + nXDst) * bpp];
                   WriteColor(pDst, DstFormat, color);
                  BYTE* pDstLine = &pDstData[(y + nYDst) * nDstStep + nXDst * bpp];
memcpy(pDstLine, pFirstDstLineXOffset, nWidth * bpp);
 ncpy.asm 中 X 反汇编 rdpgfx_main.c
891 ; Move alignment bytes.
                                   rdpgfx_codec.c
                                                                      debug_heap.cpp
                                                                                                        @ B & & %
     已引发异常
                                                                                                        诊断会话: 2 秒 (选择了 1 毫秒)
                                                                                                        ▲ 进程内存 (MB) ▼ 快照 ● 专用字节
                             ▲ 异常设置
                                                                                                        摘要 事件 内存使用率 CPU 使用率
                                  To Cam | Security | 11 z jump around to 2nd loop | going into 1st loop, ok to store deferred bytes | Security | threshold defined by cpu disp.c
                                                                                                        内存使用率
|动窗口
|名称
 index pdu pdu.fillRectCount
                 0 unsigned short (surfaceId=0 fillPixeI={B=0 \0' G=0 \0' R=0 \... RDPGFX_SO...
                                                         gin * drdynvc, _wStream * s) 行 1228
```