Version

```
https://github.com/FreeRDP/FreeRDP/commit/aa07efeb18ac9eb8df3ce8410b116b8c4b080270
```

cliprdr_server_receive_capabilities

Link

```
https://github.com/FreeRDP/FreeRDP/blob/aa07efeb18ac9eb8df3ce8410b116b8c4b080270/channels/cliprdr/server/cliprdr_main.c#L489
```

Out of Bound Read

`cliprdr_server_receive_capabilities` read data from `s` without check it's size.

```
static UINT cliprdr_server_receive_capabilities(CliprdrServerContext* context,
wStream* s,
                                                const CLIPRDR_HEADER* header)
{

    // read data without check size
    Stream_Read_UINT16(s, capabilities.cCapabilitiesSets); /* cCapabilitiesSets
(2 bytes) */
    Stream_Seek_UINT16(s);
```

double free

if `cap_sets_size` is `0`, `realloc` will **free(capabilities.capabilitySets)**, and return **NULL**, then it could **free(capabilities.capabilitySets)** again.

```
static UINT cliprdr_server_receive_capabilities(CliprdrServerContext* context,
wStream* s,
                                                const CLIPRDR_HEADER* header)
{

    for (index = 0; index < capabilities.cCapabilitiesSets; index++)
    {
        Stream_Read_UINT16(s, capabilitySetType);   /* capabilitySetType (2
bytes) */
        Stream_Read_UINT16(s, capabilitySetLength); /* capabilitySetLength (2
bytes) */
```

```
        cap_sets_size += capabilitySetLength;

        // if cap_sets_size=0, realloc --> free(capabilities.capabilitySets)
        tmp = realloc(capabilities.capabilitySets, cap_sets_size);
        if (tmp == NULL)
        {
            // free again
            free(capabilities.capabilitySets);
            return CHANNEL_RC_NO_MEMORY;
        }
```

OOB

cliprdr_read_format_list

link

```
https://github.com/FreeRDP/FreeRDP/blob/aa07efeb18ac9eb8df3ce8410b116b8c4b080270
/channels/cliprdr/cliprdr_common.c#L442
```

if `dataLen < 4` , it could lead **out of bounds read**

```
        while (dataLen)
        {
            Stream_Read_UINT32(s, formats[index].formatId); /* formatId (4
bytes) */
            dataLen -= 4;

            formats[index].formatName = NULL;
```

ntlm_read_NegotiateMessage

link

```
https://github.com/FreeRDP/FreeRDP/blob/master/winpr/libwinpr/sspi/NTLM/ntlm_mes
sage.c#L222
```

ntlm_read_AuthenticateMessage

the length of `snt` is control

```
https://github.com/FreeRDP/FreeRDP/blob/master/winpr/libwinpr/sspi/NTLM/ntlm_mes
sage.c#L812
```

ntlm_read_ntlm_v2_response

ntlm_read_ntlm_v2_response read data without check it's size
https://github.com/FreeRDP/FreeRDP/blob/master/winpr/libwinpr/sspi/NTLM/ntlm_compute.c#L166

parallel_process_irp_write

Here are some similar questions, I have introduced one of them as an example

```
static UINT parallel_process_irp_write(PARALLEL_DEVICE* parallel, IRP* irp)
{
    // read data without check stream size.
    Stream_Read_UINT32(irp->input, Length);
    Stream_Read_UINT64(irp->input, Offset);
    Stream_Seek(irp->input, 20); /* Padding */
    len = Length;
```

other similar function list

```
parallel_process_irp_read
parallel_process_irp_create
printer_process_irp_write
rdpei_recv_pdu
rdpei_recv_sc_ready_pdu
nego_process_negotiation_failure
ntlm_read_NegotiateMessage
```

int overflow

msusb_msconfig_read

link

```
https://github.com/FreeRDP/FreeRDP/blob/aa07efeb18ac9eb8df3ce8410b116b8c4b080270
/channels/urbdrc/common/msusb.c#L320
```

```
MSUSB_CONFIG_DESCRIPTOR* msusb_msconfig_read(wStream* s, UINT32 NumInterfaces)
{
    MSUSB_CONFIG_DESCRIPTOR* MsConfig;
    BYTE lenConfiguration, typeConfiguration;

    // NumInterfaces is uint32, and is control by network data.
    if (Stream_GetRemainingCapacity(s) < 6 + NumInterfaces * 2)
        return NULL;

    MsConfig = msusb_msconfig_new();

    if (!MsConfig)
```

if `NumInterfaces` is large enough, it could lead **int overflow**, and bypass the check, then lead **out of bound write later**.

progressive_wb_read_region_header

link

```
https://github.com/FreeRDP/FreeRDP/blob/aa07efeb18ac9eb8df3ce8410b116b8c4b080270
/libfreerdp/codec/progressive.c#L1990
```

tileDataSize is read from net data, and it's max value is `0xffffffff`, which could lead **int overflow**, and then bypass this check.

```
    offset += region->tileDataSize;
    if (len < offset)
    {

        return -1024;
    }
```

rdg_receive_packet

```
https://github.com/FreeRDP/FreeRDP/blob/master/libfreerdp/core/gateway/rdg.c#L30
7
```

if `packetLength<header`, then could lead **int overflow**.

```
static wStream* rdg_receive_packet(rdpRdg* rdg)
{
    wStream* s;
    const size_t header = sizeof(RdgPacketHeader);
    size_t packetLength;
    s = Stream_New(NULL, 1024);

    // first read header from network
    if (!rdg_read_all(rdg->tlsOut, Stream_Buffer(s), header))


    Stream_Seek(s, 4);
    Stream_Read_UINT32(s, packetLength);

    // if packetLenth < header coud bypass the check
    if ((packetLength > INT_MAX) || !Stream_EnsureCapacity(s, packetLength))
    {
        Stream_Free(s, TRUE);
        return NULL;
    }

    // pakcetLength-header could <0
    if (!rdg_read_all(rdg->tlsOut, Stream_Buffer(s) + header, (int)packetLength
- (int)header))
```

wts_read_drdynvc_data_first

```
https://github.com/FreeRDP/FreeRDP/blob/master/libfreerdp/core/server.
c#L193

static BOOL wts_read_drdynvc_data_first(rdpPeerChannel* channel, wStream* s, int
cbLen,
                                        UINT32 length)
{
    // read from steam
    value = wts_read_variable_uint(s, cbLen, &channel->dvc_total_length);

    if (length > channel->dvc_total_length)
        return FALSE;
    if (!Stream_EnsureRemainingCapacity(channel->receiveData, (int)channel-
>dvc_total_length))
        return FALSE;

    Stream_Write(channel->receiveData, Stream_Pointer(s), length);
    return TRUE;
```

if **dvc_total_length=0xffffffff**, then `Stream_EnsureRemainingCapacity` could overflow , and skip **alloc new memory** for `channel->receiveData`.

```
BOOL Stream_EnsureRemainingCapacity(wStream* s, size_t size)
{
    if (Stream_GetPosition(s) + size > Stream_Capacity(s)) // could int overflow
        return Stream_EnsureCapacity(s, Stream_Capacity(s) + size);
    return TRUE;
}
```

Then if `length > Steam_Capacity(channel->receiveData)` , it could oob write when Steam_Write.

```
    Stream_Write(channel->receiveData, Stream_Pointer(s), length);
    return TRUE;
```