An NCC Group Publication

USB attacks need physical access right? Not any more...

Prepared by:
Andy Davis
Research Director
andy.davis ‘at’ nccgroup ‘dot’ com

Contents

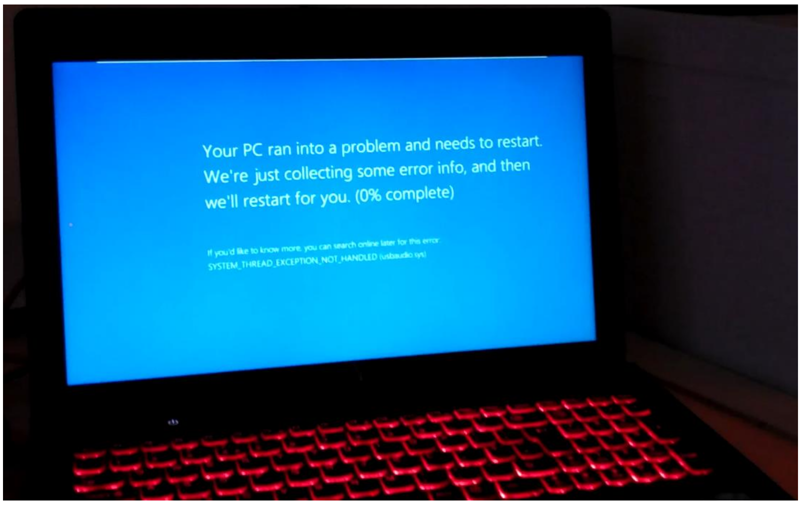
List of Figures and Tables

1 Introduction

Historically USB bugs have required physical access so that a rogue device can be inserted into the target system to trigger a vulnerability by supplying malicious data, often within a USB protocol descriptor. This paper provides step-by-step instructions, showing how to remotely trigger a Windows-based USB bug by using a combination of open source hardware, open source software and new functionality provided by Microsoft RDP server.

2 The bug and how it can be triggered

The USB bug that will be used for the demonstration resides in an audio driver (*usbaudio.sys*) and runs in kernel mode on all current versions of Microsoft Windows. When it is triggered a Bugcheck (or BSOD) occurs; however, Microsoft has confirmed that it is a Denial of Service bug that does not result in code execution and therefore, they do not consider it to be a security issue.



Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (0% complete)

If you'd like to know more, you can search online later for this error: SYSTEM_THREAD_EXCEPTION_NOT_HANDLED (usbaudio.sys)

Figure 1: Bugcheck triggered by bug in usbaudio.sys

The bug can be triggered using *umap*[*] (**Note:** ensure you have the latest *testcases.py*) as follows:

```
$ sudo python3 umap.py -P /dev/ttyUSB0 -s 01:01:00:C:4
---------------------------------------

 _   _ _ __ ___   __ _ _ __
| | | | '_ ` _ \ / _` | '_ \
| |_| | | | | | | (_| | |_) |
 \__,_|_| |_| |_|\__,_| .__/
                      |_|


The USB host assessment tool
Andy Davis, NCC Group 2013
Version: 1.0

Based on Facedancer by Travis Goodspeed

For help type: umap.py -h
---------------------------------------


Fuzzing:
2014/02/18 12:43:34 Class-specific data...
 Audio class: 0004 - CSInterface1_bInCollection_null
```

[*] https://github.com/nccgroup/umap

3 USB Redirection via RDP

For many years the remoting of different USB devices has been possible via RDP, using a method known as “High-Level” redirection. The support for different USB device types has increased steadily as different device-specific remoting techniques have been added to the RDP protocol; examples are listed below:

- Easy Print
- Drive Redirection
- Smart Card Redirection
- Plug-and-Play Device Redirection
- Input Redirection
- Audio Redirection
- Port Redirection

3.1 Problems with remote exploitation via High-Level redirection

The fundamental problem with trying to exploit a USB bug via High-Level redirection is that the driver used for the USB device resides on the client, not on the server and therefore, an exploitation attempt would look something like Figures 2 to 4.

Firstly, the RDP client is configured to use High-Level redirection for the USB device associated with the attack then a connection is established between the RDP client and RDP server:

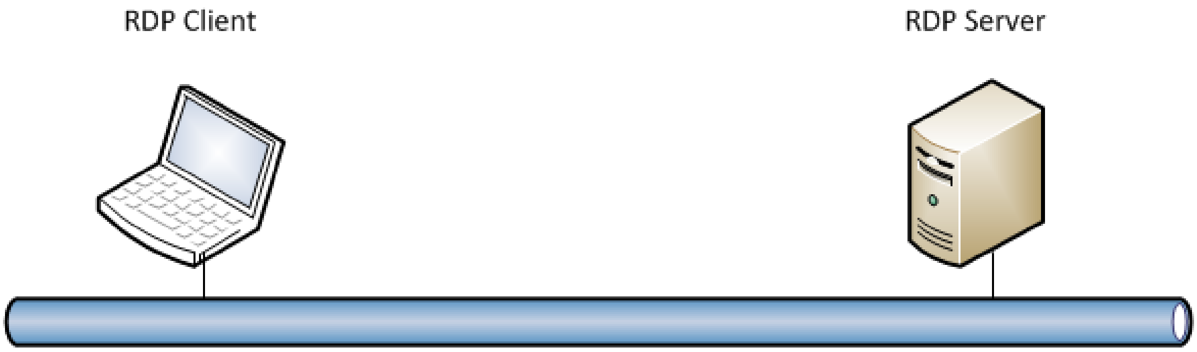


Figure 2: RDP client connected to an RDP server

Next, the rogue USB device (that will trigger the bug) is inserted into the RDP client machine:

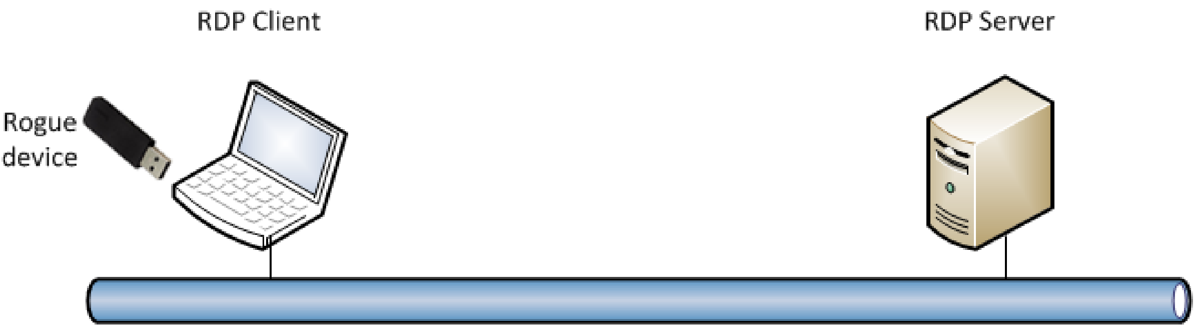


Figure 3: A rogue USB device is connected to the client

However, because the driver used in High-Level USB redirection resides on the RDP client machine, the bug is triggered on the client rather than the server (if the client machine is not vulnerable to the bug being tested then nothing will happen):

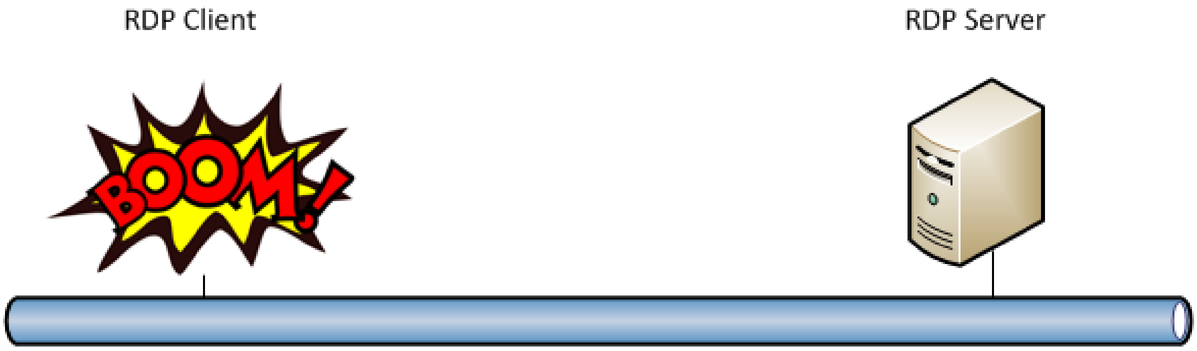


Figure 4: The bug is triggered on the RDP client rather than the RDP server

4 RemoteFX USB Redirection

However, there is another, more recent USB redirection technology that was introduced with Windows Server 2012, called “RemoteFX USB redirection”. Table 1 lists the differences between RemoteFX and High-Level USB redirection technologies and as highlighted within the table, the most important change from the perspective of remote exploitation is that with RemoteFX, the USB driver used is the one installed on the RDP server.

RemoteFX USB Redirection	RDP High-Level Device Redirection
Does not require drivers on the client	Requires drivers for the device to be installed on the client
Requires device driver to be installed on the server	Generally does not require drivers on the server
Uses on redirection method for many types of devices	Uses a specific, unique method for each type of device being redirected
Forwards URBs to and from the device over the DRP connection	Exposes high-level device functionality in the remote session by using an optimised protocol for the device type
Enables only one session to use a device at a given time; the local client cannot use the device while an RDP session is using it	Enables any number of sessions to access the device simultaneously, including the local client
Is optimised for the LAN, like the rest of RemoteFX	Works with both LAN and WAN

Table 1: A comparison between RemoteFX and High-Level USB redirection (from blogs.msdn.com)

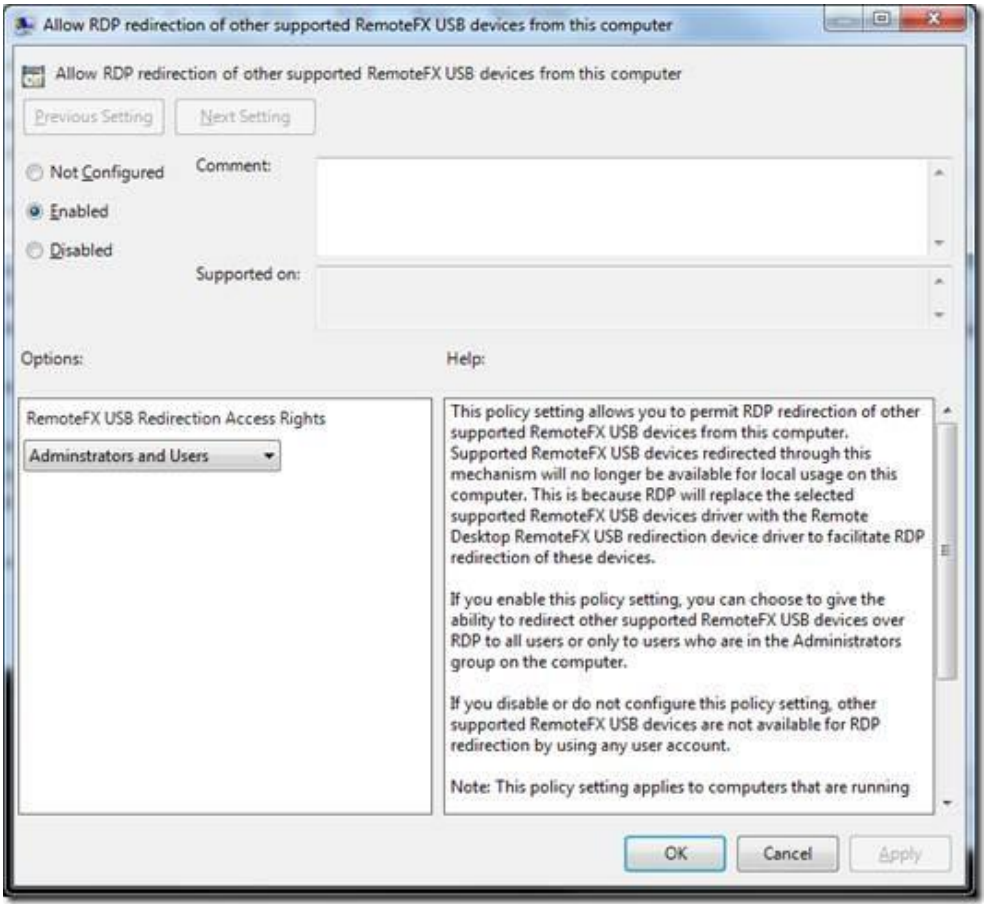
4.1 Enable RemoteFX on the client

In order to use RemoteFX USB redirection, the RDP client needs to be configured to do so, by setting the following option in the local Windows Group Policy settings:

Group Policy name: Allow RDP redirection of other supported RemoteFX USB devices from this computer

Group Policy setting:

```
Computer Configuration\Administrative Templates\Windows Components\Remote Desktop
Services\Remote Desktop Connection Client\RemoteFX USB Device Redirection
```



Allow RDP redirection of other supported RemoteFX USB devices from this computer

Allow RDP redirection of other supported RemoteFX USB devices from this computer

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on:

Options:

RemoteFX USB Redirection Access Rights

Adminstrators and Users

Help:

This policy setting allows you to permit RDP redirection of other supported RemoteFX USB devices from this computer. Supported RemoteFX USB devices redirected through this mechanism will no longer be available for local usage on this computer. This is because RDP will replace the selected supported RemoteFX USB devices driver with the Remote Desktop RemoteFX USB redirection device driver to facilitate RDP redirection of these devices.

If you enable this policy setting, you can choose to give the ability to redirect other supported RemoteFX USB devices over RDP to all users or only to users who are in the Administrators group on the computer.

If you disable or do not configure this policy setting, other supported RemoteFX USB devices are not available for RDP redirection by using any user account.

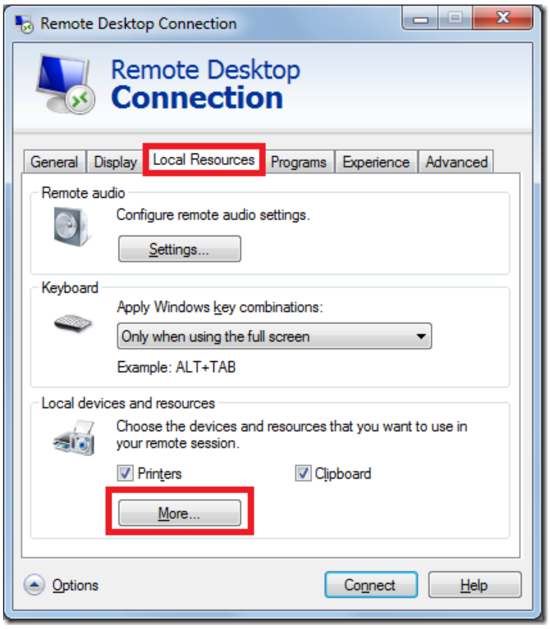
Note: This policy setting applies to computers that are running

Figure 5: Local Windows policy setting to enable RemoteFX USB redirection

Once the setting has been changed, the following command needs to be issued and then the client machine rebooted:

```
C:\> gpupdate /force
```

Once the machine has rebooted and the RDP client started, select “Local Resources”, as shown in Figure 6 and a new checkbox will have appeared, called “Other supported RemoteFX USB devices”, as shown in Figure 7. This confirms that RemoteFX USB redirection has been successfully configured on the RDP client.



Remote Desktop Connection

Remote Desktop Connection

General | Display | Local Resources | Programs | Experience | Advanced

Remote audio

Configure remote audio settings.

Settings...

Keyboard

Apply Windows key combinations:

Only when using the full screen

Example: ALT+TAB

Local devices and resources

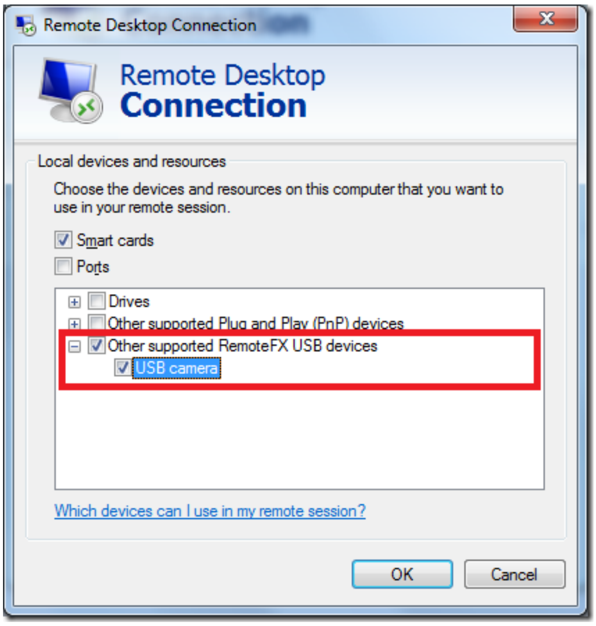
Choose the devices and resources that you want to use in your remote session.

☑ Printers ☑ Clipboard

More...

Options | Connect | Help

Figure 6: RDP client Local Resource settings



Remote Desktop Connection

Remote Desktop Connection

Local devices and resources

Choose the devices and resources on this computer that you want to use in your remote session.

☑ Smart cards

☐ Ports

- ☐ Drives
- ☐ Other supported Plug and Play (PnP) devices
- ☑ Other supported RemoteFX USB devices
  - ☑ USB camera

Which devices can I use in my remote session?

OK | Cancel

Figure 7: RDP client RemoteFX USB device settings

4.2 Remote FX USB redirection attack

Now we are ready to launch the USB attack against a remote Windows 2012 server running an RDP server.

First, select the “Other supported RemoteFX USB devices” checkbox within the RDP client settings and then establish an RDP session with the remote machine (Figure 8).

Note: The attacker still needs to authenticate with the remote machine using a valid user account; however, the privilege level of this user is irrelevant (the built-in “Guest” user can successfully be used, if it is enabled).

RDP Client
RDP Server

Figure 8: RDP client configured to use RemoteFX connected to a Windows 2012 RDP server

Next, a legitimate USB device is inserted into the RDP client machine to establish the redirection associated with that device (Figure 9). This can be achieved using *umap* as follows:

```
$ sudo python3 umap.py -P /dev/ttyUSB0 -e 01:01:00
---------------------------------------
 _   _ _ __ ___   __ _ _ __
| | | | '_ ` _ \ / _` | '_ \
| |_| | | | | | | (_| | |_) |
 \__,_|_| |_| |_|\__,_| .__/
                      |_|

The USB host assessment tool
Andy Davis, NCC Group 2013
Version: 1.0

Based on Facedancer by Travis Goodspeed

For help type: umap.py -h
---------------------------------------

Emulating 01:01:00 - Audio : Audio control : PR Protocol undefined
Facedancer reset
GoodFET monitor initialized
MAXUSB initialized
MAXUSB enabled
MAXUSB revision 19
MAXUSB connected device USB audio device
USB audio device received request dir=1, type=0, rec=0, r=6, v=256, i=0, l=64
USB audio device received GET_DESCRIPTOR req 1, index 0, language 0x0000, length
64
MAXUSB wrote 12 01 00 02 00 00 00 40 1e 04 02 04 00 01 01 02 03 01 to endpoint 0
USB audio device received request dir=0, type=0, rec=0, r=5, v=7, i=0, l=0
USB audio device received SET_ADDRESS request for address 7
USB audio device received request dir=1, type=0, rec=0, r=6, v=256, i=0, l=18
USB audio device received GET_DESCRIPTOR req 1, index 0, language 0x0000, length
18
MAXUSB wrote 12 01 00 02 00 00 00 40 1e 04 02 04 00 01 01 02 03 01 to endpoint 0
USB audio device received request dir=1, type=0, rec=0, r=6, v=512, i=0, l=255
USB audio device received GET_DESCRIPTOR req 2, index 0, language 0x0000, length
255
MAXUSB wrote 81 03 04 00 02 to endpoint 0
<Truncated for brevity>
```

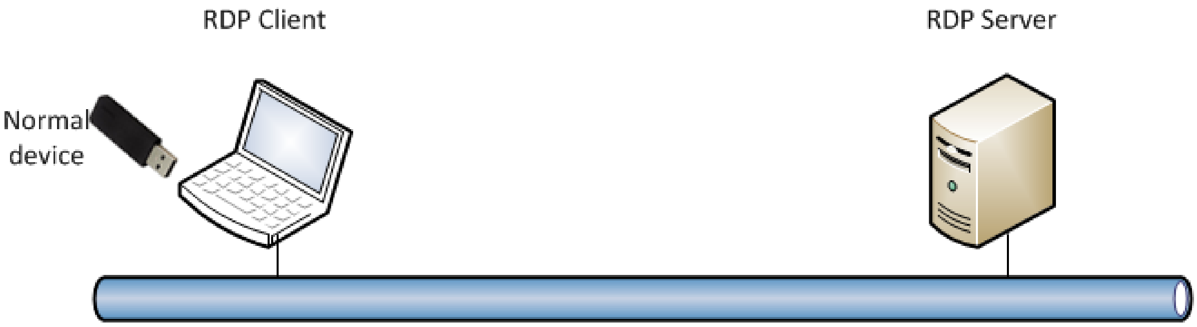


Figure 9: Legitimate USB device inserted into the RDP client machine

Within the RDP session a new “Devices” icon is displayed (Figure 10), which when clicked will display the device type that can be remoted, which in this instance is a “Creative HS-720 Headset” (Figure 11).

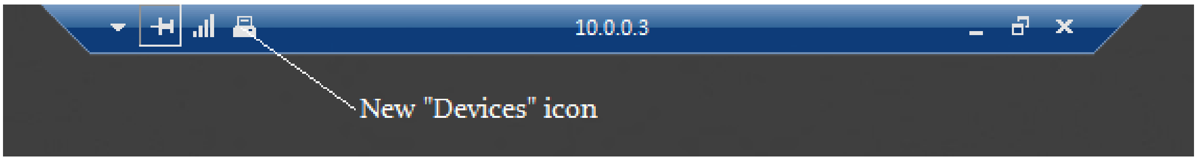
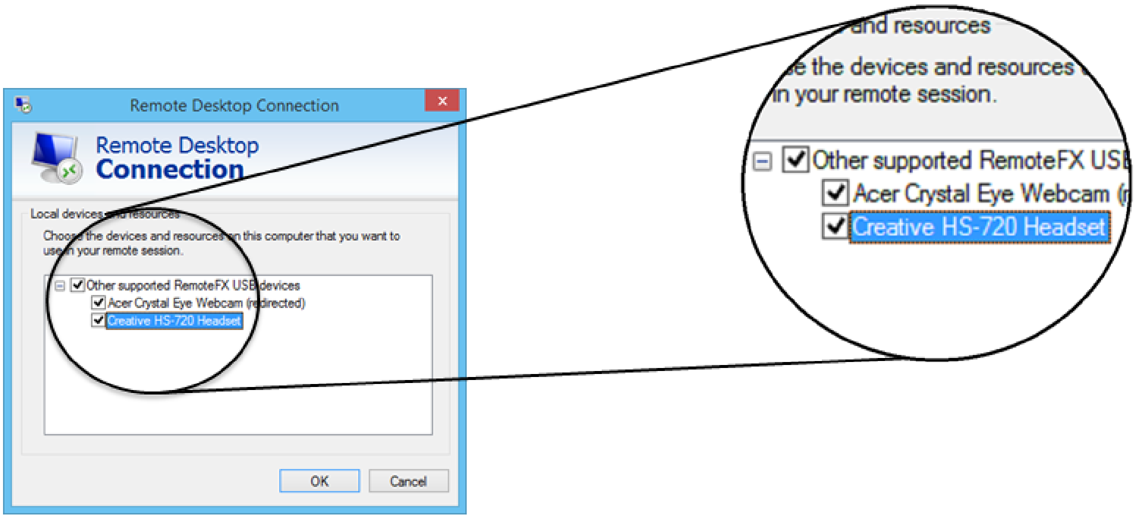


Figure 10: The RDP session “Devices” icon



Remote Desktop Connection

Remote Desktop Connection

Local devices and resources

Choose the devices and resources on this computer that you want to use in your remote session.

☑ Other supported RemoteFX USB devices
- ☑ Acer Crystal Eye Webcam (redirected)
- ☑ Creative HS-720 Headset

Figure 11: The USB devices that can be redirected via RemoteFX USB redirection

Quickly select this checkbox and click “OK” to enable future remoting of this device (before the device emulation in *umap* times out). Next, (virtually) insert the rogue device by issuing the following *umap* command line:

```
$ sudo python3 umap.py -P /dev/ttyUSB0 -s 01:01:00:C:4
```

This will now use the previously selected USB redirection for the device during the session (Figure 12) and trigger the bug on the remote server.

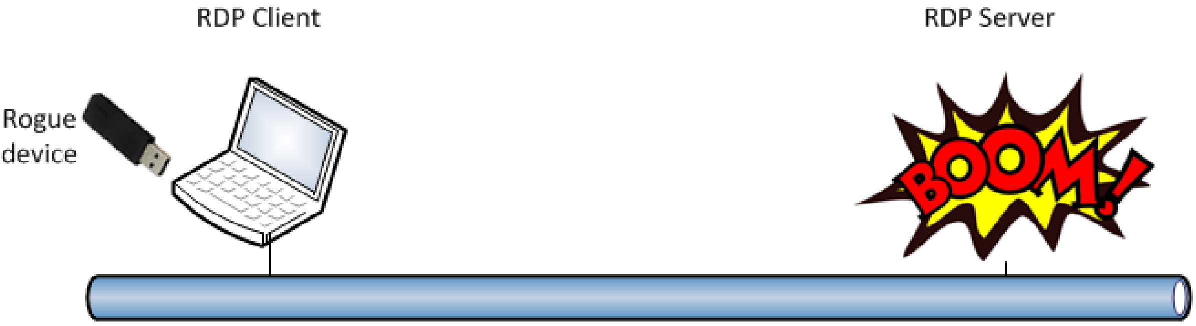


Figure 12: The rogue device is now inserted and will be redirected via RemoteFX USB redirection

5 How can you reduce the risks?

There are a number of steps you can take to reduce the risks associated with this type of attack; these are detailed below.

5.1 Disable RemoteFX on servers

If RemoteFX is not required on the server, turn it off using Group Policy.

Group Policy name: Configure RemoteFX

Group Policy setting:

```
Computer Configuration\Administrative Templates\Windows Components\Remote Desktop
Services\Remote Desktop Session Host \Remote Session Environment
```

5.2 Disable RemoteFX on clients

Disable RemoteFX USB redirection on clients using Group Policy.

Group Policy name: Allow RDP redirection of other supported RemoteFX USB devices from this computer

Group Policy setting:

```
Computer Configuration\Administrative Templates\Windows Components\Remote Desktop
Services\Remote Desktop Connection Client\RemoteFX USB Redirection
```

Group Policy name: Do not allow supported Plug and Play device redirection

Group Policy setting:

```
Computer Configuration\Administrative Templates\Windows Components\Remote Desktop
Services\Remote Desktop Session Host \Device and Resource Redirection
```

5.3 Use more granular RemoteFX security controls

Information about how more granular control can be exercised over RemoteFX USB redirection is provided by Microsoft on the MSDN[†] site.

5.4 Take “local” USB vulnerabilities more seriously

As it has now been demonstrated that the potential impact associated with USB bugs has increased, be more cautious of “local” USB vulnerabilities and ensure that any security patches released are appropriately installed.

6 Conclusions and implications for future USB bugs

This paper has shown that USB bugs are no longer just a threat associated with physical access to machines. The recent addition of more fully featured remoting technologies has extended the reach of USB attacks and with Microsoft Windows, has increased the exposure to the kernel via attacks against USB drivers over the network. Therefore, appropriate steps, as described above, should be taken to secure server infrastructures where RemoteFX USB redirection could be deployed.

[†] http://blogs.msdn.com/b/rds/archive/2010/06/10/introducing-microsoft-remotefx-usb-redirection-part-2.aspx