nccgroup
freedom from doubt

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

Andy Davis, Research Director NCC Group

nccgroup
freedom from doubt

Agenda

- The problem with USB bugs
- The USB bugs I've found and how I found them
- **Demo:** iOS 7 bug
- A contrived example of a remote USB bug via Bluetooth
- **Demo:** Windows *hidparse.sys* bug
- USB redirection via RDP
- **Demo:** Windows 2012 *usbaudio.sys* bug (triggered remotely)
- The implications for future USB bugs
- Mitigation strategies to reduce the risks
- Conclusions and further research

The problem with USB bugs

- Physical access required
- Vendors aren’t really interested
- Local bugs have their place, but have a limited impact

Image: www.norcalblogs.com

nccgroup
freedom from doubt

Recent high profile USB host bugs

- **CVE-2011-2295:** Oracle Sun Solaris USB Local Buffer Overflow Vulnerability
- **CVE-2012-3723:** Apple Mac OS X USB Hub Descriptor bNbrPorts Heap overflow
- **MS13-027:** The Windows 8 RNDIS kernel pool overflow
- **CVE-2013-3200:** Microsoft Windows USB Descriptor Handling Local Privilege Escalation

nccgroup
freedom from doubt

How I first started finding USB bugs (2011)

- Arduino microcontroller
- Fuzzer written in C++
- Only emulates USB HID devices
- Only allows semi-automated fuzzing
- Has found bugs in:
  - Windows 7
  - Windows XP
  - OS X
- Limitations – not really fast enough to emulate most USB devices

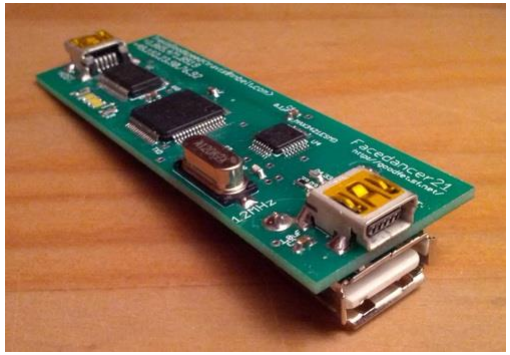
USB fuzzer – the next generation (2012)

- Dedicated USB test equipment hardware
- USB capture and playback
- Emulates any USB host or device
- Understands and analyses the different USB device classes
- Uses a scripting language to generate USB traffic
- Costs approx. USD1200 (plus specific class analysis options)
- Limitations – doesn't have a software API to control it

Image: www.mqp.com

How I find them now (2013…)

- Facedancer and umap



```
$ sudo python umap.py -P /dev/ttyUSB0 -i

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

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
----------------------------------------

01:01:00 - Audio : Audio control : PR Protocol undefined
01:02:00 - Audio : Audio streaming : PR Protocol undefined
02:02:01 - CDC Control : Abstract Control Model : AT commands V.250
02:03:ff - CDC Control : Telephone Control Model : Vendor specific
02:06:00 - CDC Control : Ethernet Networking Control Model : No class-specific protocol required
03:00:00 - Human Interface Device : No subclass : None
**SUPPORTED**
06:01:01 - Image : Still image capture device : Bulk-only protocol
07:01:02 - Printer : Default : Bidirectional interface
08:06:50 - Mass Storage : SCSI : BBB
 **SUPPORTED**
09:00:00 - Hub : Default : Default
 **SUPPORTED**
0a:00:00 - CDC Data : Default : Default
0b:00:00 - Smart Card : Default : Default
```

- https://github.com/nccgroup/umap

Image: http://blog.j-michel.org/

What has umap found most recently?

- CVE-2014-1287 : iOS 7 arbitrary code execution in kernel mode

Image: ipadinministry.revguy.net

nccgroup
freedom from doubt

Demo: iOS 7 bug

Image: us.cdn4.123rf.com

nccgroup
freedom from doubt

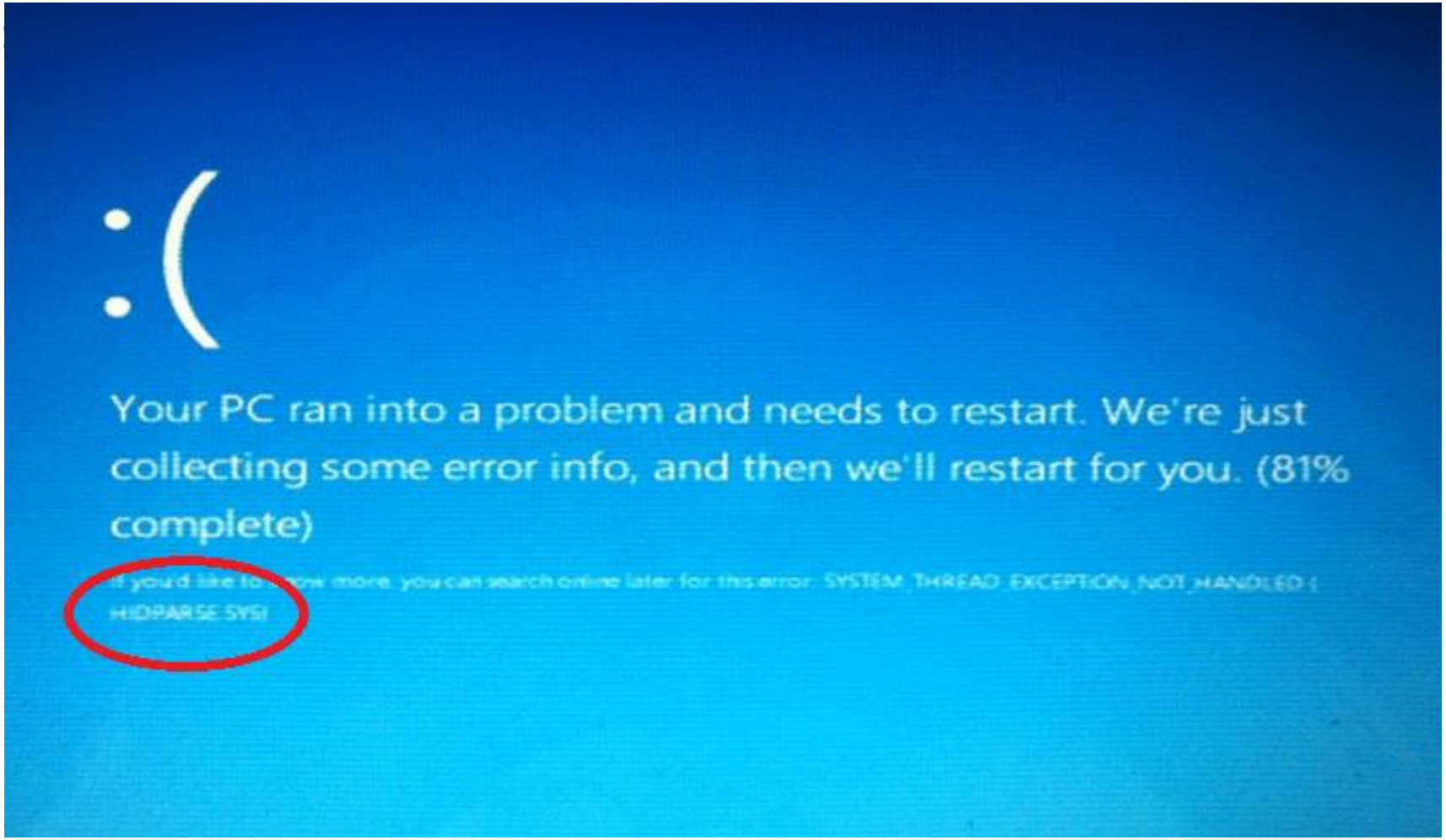
Contrived example of remote USB bug

- Discovered a null-pointer dereference in *hidparse.sys* in 2011
- Microsoft not interested – not really a security issue
- Triggered by inserting a malicious device
- HID report descriptor is parsed during enumeration phase
- Blue screen occurs

Contrived example of remote USB bug

nccgroup
freedom from doubt

Windows (all versions) – not patched



nccgroup
freedom from doubt

What else uses HID report descriptors?

- Bluetooth keyboards!
- During pairing, report descriptor is parsed
- Very tenuous, as this isn’t really a USB bug it’s a HID parsing bug
- First attempt to move away from direct physical access

nccgroup
freedom from doubt

I have a spare Arduino…

Small, portable, programmable USB device to trigger USB vulnerabilities:

Demo: *hidparse.sys* bug via Bluetooth

Image from: us.cdn4.123rf.com

USB Redirection via RDP

Numerous legacy “High-level” redirection capabilities:

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

nccgroup
freedom from doubt

Problem with High-level USB redirection

Problem with High-level USB redirection

nccgroup
freedom from doubt

Problem with High-level USB redirection

Enter “RemoteFX USB Redirection”…

Image: www.norcalblogs.com

RemoteFX USB Redirection

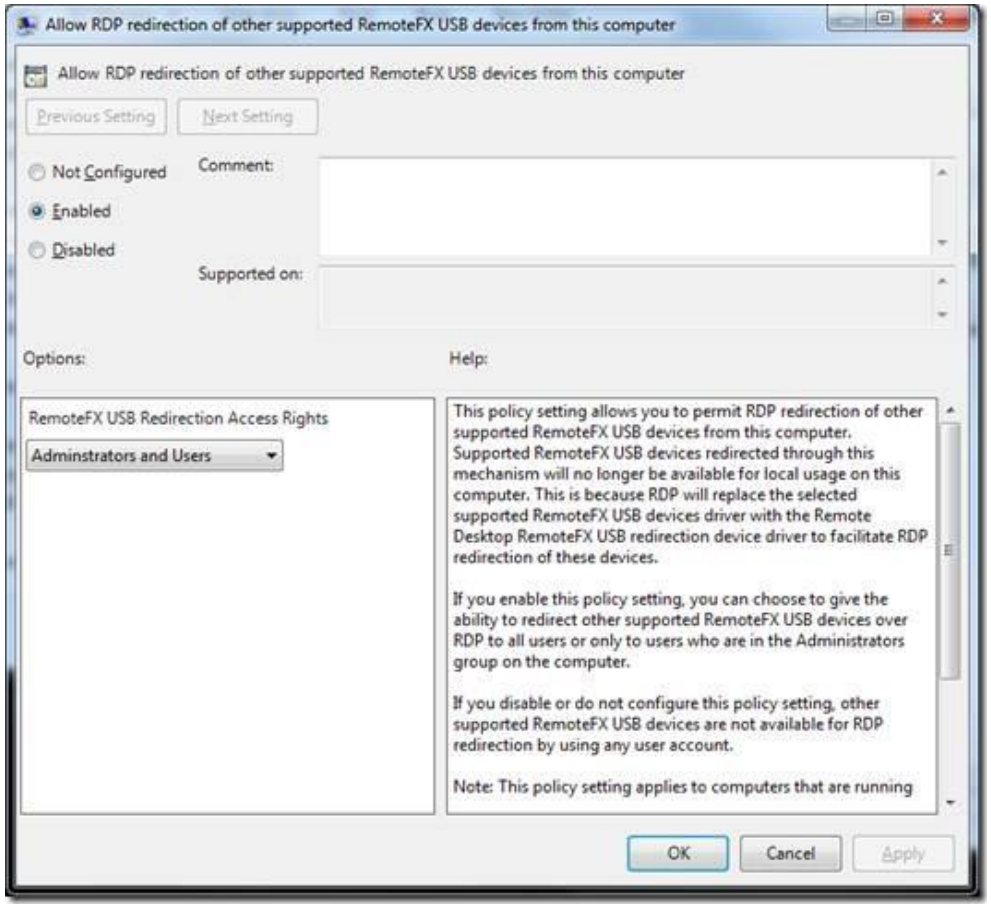
nccgroup
freedom from doubt

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 the device driver to be installed on the server	Generally does not require drivers on the server
Uses one 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 RDP connection	Exposes high-level device functionality in the remote session by using an optimized 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 optimized for the LAN, like the rest of RemoteFX	Works with both LAN and WAN

Enable RemoteFX on the client

nccgroup
freedom from doubt

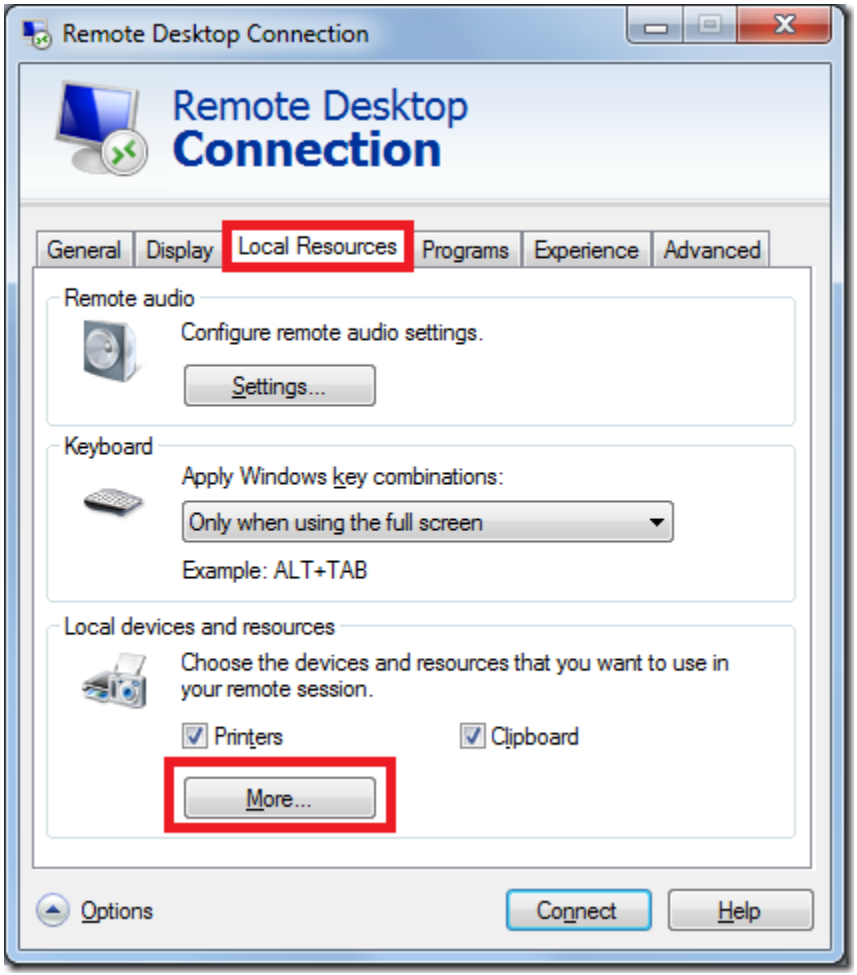
Computer Configuration\Administrative Templates\Windows Components
\Remote Desktop Services\Remote Desktop Connection Client\
RemoteFX USB Device Redirection

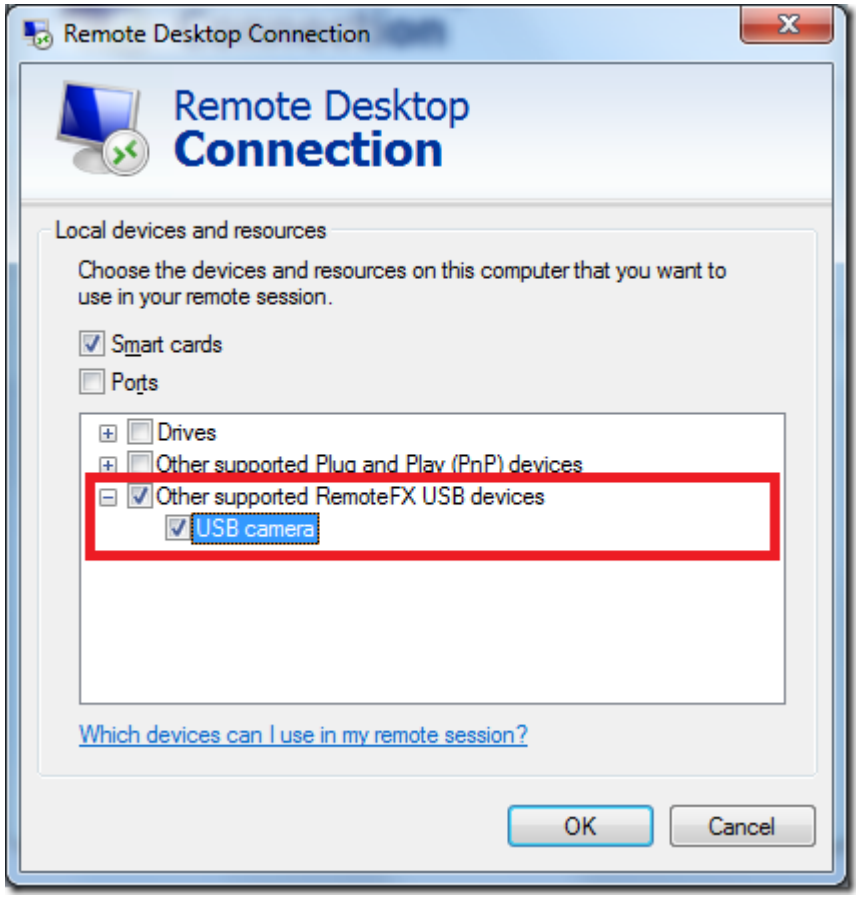


```
C:\> gpupdate /force
```

Enable RemoteFX on the client

nccgroup
freedom from doubt

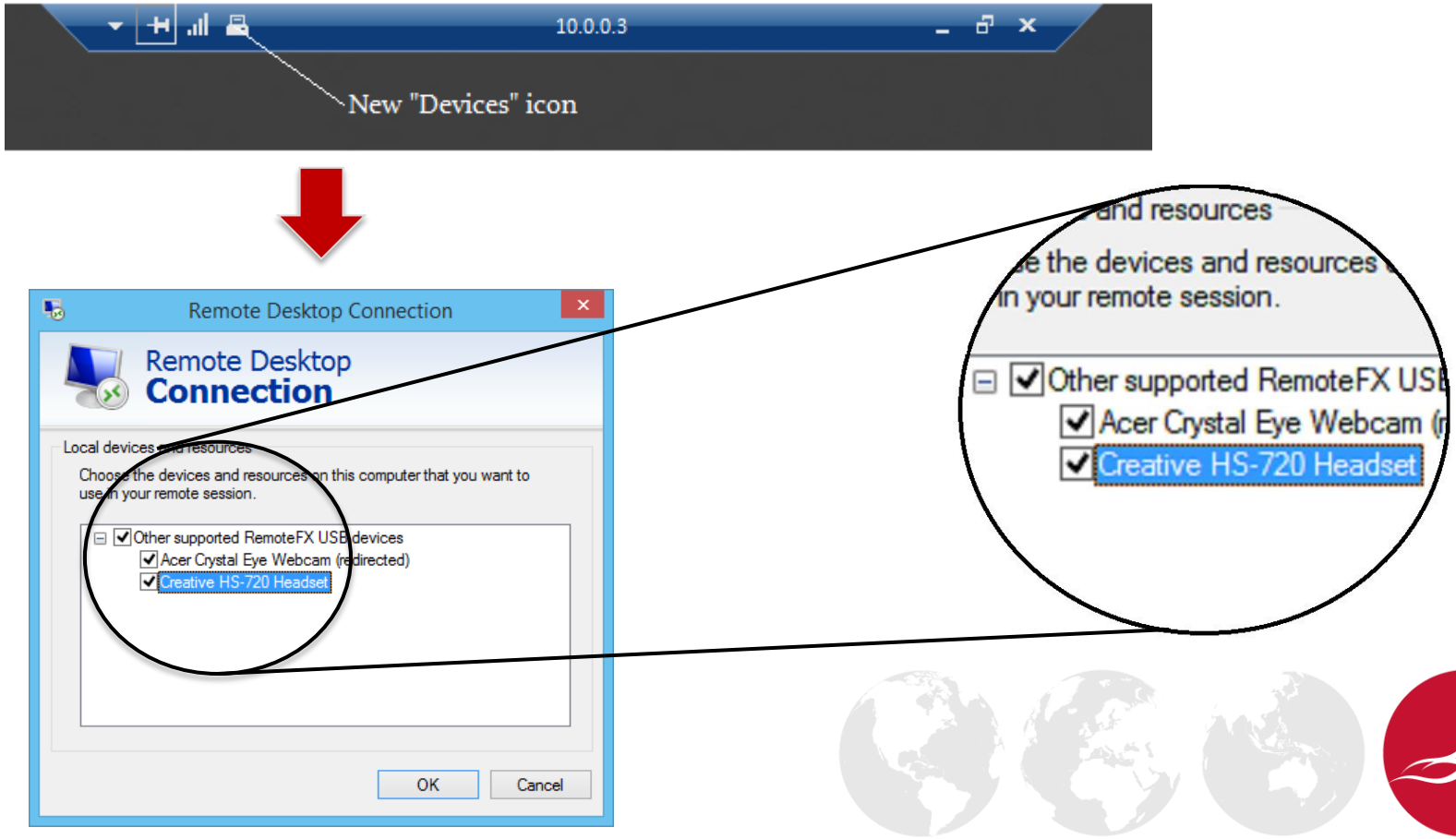




Remote FX USB redirection attack

Remote FX USB redirection attack

Remote FX USB redirection attack



Remote FX USB redirection attack

Remote FX USB redirection attack

Demo: *usbaudio.sys* bug via RDP

Image: us.cdn4.123rf.com

The implications for future USB bugs

- Windows USB bugs no longer need local physical access
- Remote exposure of the Windows kernel has been increased
- What were local DoS bugs can now remotely “blue-screen” a server
- May apply to other (non-Windows) remoting technologies

Image: montanaacquisitions.co.uk

How can you reduce the risks?

- If RemoteFX is not required on the server, turn it off
- If RemoteFX is required specify GUIDs of authorised USB devices
- Do not enable RemoteFX USB remoting on clients
- Minimise the use of USB “High-level” remoting via RDP
- Be more cautious of “local” vulnerabilities and apply the patches

Image: www.tradeportindia.com

Conclusions and further research

- Physical access is no longer a requirement to trigger Windows USB bugs
- RemoteFX USB remoting has exposed more of the Windows kernel to attackers
- Need to investigate other remoting technologies e.g. Citrix
- The Internet of Things is full of USB possibilities ☺

Questions?

Andy Davis, Research Director NCC Group

andy.davis ‘at’ nccgroup ‘dot’ com

nccgroup
freedom from doubt

UK Offices

Manchester - Head Office

Cheltenham

Edinburgh

Leatherhead

London

Thame

North American Offices

San Francisco

Atlanta

New York

Seattle

Australian Offices

Sydney

European Offices

Amsterdam - Netherlands

Munich – Germany

Zurich - Switzerland