nccgroup
freedom from doubt

An NCC Group Publication

What the HEC? Security implications of HDMI Ethernet Channel and other related protocols

Prepared by:
Andy Davis
Research Director
andy.davis 'at' nccgroup.com

Contents

1 List of Figures and Tables

2 Introduction

VGA is dead (or at least dying), long live HDMI. The VGA socket on the back of your PC is 25 years old and with new PCs and laptops we're starting to see a change – to HDMI, but under the hood this brings many other capabilities than just displaying video.

This paper discusses the various communications protocols that exist within HDMI to provide a whole host of plug-and-play functionality and the security impact of exposing these technologies to a corporate environment. Also discussed is a tool for assessing some of these protocols to identify security vulnerabilities within software that processes them.

3 Why research video interfaces?

At the time of publication there is very little[1][2] publicly available research investigating the security of video interface technologies and therefore, this paper intends to raise awareness of the potential security issues and scenarios which are likely to arise as the protocols start to be incorporated into mainstream PC-based operating systems.

Conventional wisdom states that a video interface is solely an output; however, this has not strictly been the case for many years. Clearly, video (and often audio) are outputs, but since the early days of VGA[3] (Video Graphics Array), the EDID[4] (Extended Display Identification Data) protocol has been used to provide information about display capabilities to the video interface cards of hosts to which displays are attached. EDID is still used in HDMI, but a number of other new bi-directional protocols have also been added; these include: CEC[5] (Consumer Electronics Control), HEC[6] (HDMI Ethernet Channel) and HDCP[7] (High-bandwidth Digital Content Protection). CEC and HEC will be discussed extensively in this paper.

4 HDMI overview

The full HDMI (High Definition Multimedia Interface) protocol is detailed in the HDMI Specification v1.4a[8], which is available from HDMI Licensing LLC, however, it is only available to licensed HDMI adopters*. HDMI is a digital video interface technology that transmits both video and audio from a digital source device such as a Blu-ray player to a digital sink device such as a plasma TV. It transmits encrypted uncompressed digital video and audio data using TMDS[9] (Transition-Minimized Differential Signalling). It also supports Enhanced DDC[10] (Display Data Channel) for display identification and capability advertisement and a number of other new protocols, which have already been mentioned:

- CEC – used for the control of HDMI devices
- HEC – used to provide Ethernet networking capabilities to HDMI
- HDCP – used to provide authentication, encryption and key revocation services to HDMI

The CEC and HEC protocols will now be discussed in more detail along with potential security issues they can introduce.

* NCC Group is not a licensed HDMI adopter - the technical information about HDMI within this paper has been compiled from freely available sources on the Internet

5 Consumer Electronics Control

The majority of vendors who utilise HDMI technology do not use the term CEC. Instead, marketing terms are applied to make the technology sound more “friendly” or more aligned to their product lines. Trade names for CEC include:

- BRAVIA Link and BRAVIA Sync (Sony)
- VIERA Link , HDAVI Control, EZ-Sync (Panasonic)
- Anynet+ (Samsung)
- Aquos Link (Sharp)
- SimpLink (LG)
- EasyLink (Philips)

The motivation behind the CEC protocol is twofold; firstly, to enable users to control two or more HDMI devices using a single remote control and secondly, to enable devices to control each other without user-intervention.

The architecture of CEC is an inverted tree with the “root” node (display) at the top, “branch” nodes (video switches) below and connected to “leaf” nodes (devices such as Blu-ray players). This architecture is shown in Figure 1.

Figure 1: CEC architecture

The concept is that the user should only need to perform the minimal number of tasks and let the “system” do the rest e.g. increasing the volume on the TV remote would identify which connected products need to be communicated with to ensure the volume of the current video /audio source increases.

CEC is a one-wire bidirectional serial bus that uses the AV.link[11] protocol for communication. Up to ten A/V devices can be connected and the topology of a connected system is auto-discovered by the protocol, as all logical device addresses must be unique. The messages that can be sent can be either device-specific or general e.g. power control.

Some examples of CEC commands (or “Features” as they are known in CEC) are:

- One Touch Play: the device will become active source when playback starts
- System Standby: switches all connected devices to standby
- Pre-set Transfer: transfers the tuner channel setup to another TV set
- One Touch Record: start recording immediately
- Timer Programming: allow one device (e.g. a TV set or HTPC) to set the timer programming of another (e.g. a PVR,/DVR or DVD-recorder)
- System Information: checks all components for bus addresses and configuration
- Deck Control: playback control
- Tuner Control: control the tuner of another device
- OSD Display: use the OSD of the TV set to display text
- Device Menu Control: use the menus of another device
- Routing Control: control the switching of signal sources
- Remote Control Pass Through: pass through remote control commands
- Device OSD Name Transfer: transfer the preferred device names to the TV set

Some of these Features e.g. “System Standby” are sent to a device as a direct result of a user pressing a button, whereas others, such as those associated with “Routing Control” are sent between devices as a consequence of other actions being performed.

5.1 Protocol details

Bits are grouped into 10-bit header and data blocks. Both header and data blocks include 8-bits of data along with EOM and ACK bits, as shown in Figure 2.

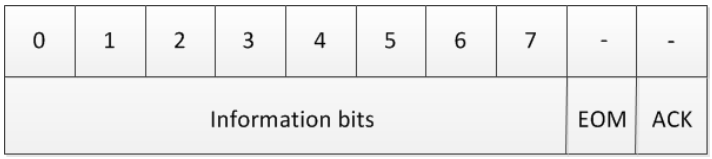

Figure 2: CEC Block layout

Figure 3 shows the structure of a CEC header.

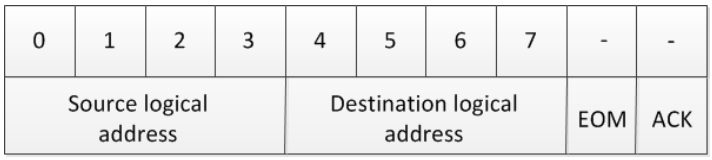

Figure 3: CEC Header

The EOM bit signals the final block in a message. A ‘0’ indicates that one or more blocks follow and a ‘1’ indicates the message is complete. When a single follower provides an ACK to an initiator, it does so by “overriding” the output from the initiator (i.e. by pulling the bus to a logical ‘0’ while the Initiator sends a “passive” logical ‘1’). Broadcast messages have special rules for handling simultaneous ACKs from multiple devices. Here, the logic is reversed and a group of followers ACK by not “overriding” the initiator (i.e. by allowing the Initiator to send a “passive” logical ‘1’).

CEC devices have both physical and logical addresses. Normally, for each hot-plug, the CEC source obtains a physical address by reading the Vendor Specific Data Block[12] within the EDID data from the sink it is attached to. A Vendor Specific Data Block contains as its first three bytes the vendor's IEEE 24-bit registration number, LSB first. For HDMI, it is always 00-0C-03 (HDMI Licensing, LLC). It is followed by two bytes, which represent the CEC source physical address, LSB first. The physical

address of each CEC device is expressed as four numbers and indicates where it is relative to the "root" display, whose address is always fixed at 0.0.0.0. For example, a source attached to input #1 of the "root" display, will have a physical address of 1.0.0.0. If it were connected to input #2, the physical address would be 2.0.0.0 etc. Each CEC device also obtains a logical address - reflecting its product type – by negotiating with other CEC devices in the system. Table 1 lists all the possible CEC logical addresses.

Address	Device
0	TV
1	Recording device 1
2	Recording device 2
3	Tuner 1
4	Playback device 1
5	Audio system
6	Tuner 2
7	Tuner 3
8	Playback device 2
9	Recording device 3
10	Tuner 4
11	Playback device 3
12	Reserved
13	Reserved
14	Free use
15	Unregistered (as initiator address) Broadcast (as destination address)

Table 1: CEC logical addresses

A device may lose its logical address within the network when it is disconnected or switched off. However, if it has the capability to remember its previous logical address, the next time it is reconnected or switched on, it can begin the polling process at its previous logical address and try each other allowable logical address in sequence before taking the unregistered address.

A message with the EOM bit set in the Header Block can be used to 'ping' other devices, to ascertain if they are powered on. This is the `<Polling Message>` and the initiator and destination addresses will be different. It is also used for allocating logical addresses: in this case the initiator and destination addresses are the same. The format of a CEC message is shown in Figure 4.

Figure 4: CEC Message

In the HDMI 1.4a standard there are 71 different CEC Features. Each Feature is represented by an opcode and depending on the Feature, followed by a number of operand bytes e.g. to put a device into standby, the "Standby" Feature (opcode 0x36) is sent with no operand bytes. Whereas, to set an on-screen display string, the "Set OSD String" Feature (opcode 0x64) is sent with operand bytes representing the string to display. Messages are always either "Directed" or "Broadcast" and logical address 0xF is always the Broadcast address.

5.2 Fuzzing CEC

As CEC is a feature rich protocol it could potentially yield some security vulnerabilities in different implementations. Interesting CEC Features from a fuzzing perspective include:

- "Vendor command" (opcode 0x89) – this is totally customisable by the vendor
- "Set OSD string" (opcode 0x64) – potential for buffer overflow or format string bugs
- "Set OSD name" (opcode 0x47) – potential for buffer overflow or format string bugs
- "CDC command" (opcode 0xF8) – the CDC protocol is covered in section 5.1

So now that we have some understanding of the CEC protocol and the type of features that can potentially be fuzzed the next step is to understand how to interface with CEC, ideally from a PC. There is an open source library that has been written for the Arduino microcontroller and Figure 5 shows the interface circuit. However, a more flexible approach is to use a USB-based CEC adapter, such as the "USB-CEC Adapter"[13] from Pulse Eight or "USB-CEC Bridge"[14] from RainShadow Tech shown in Figures 6 and 7 respectively.

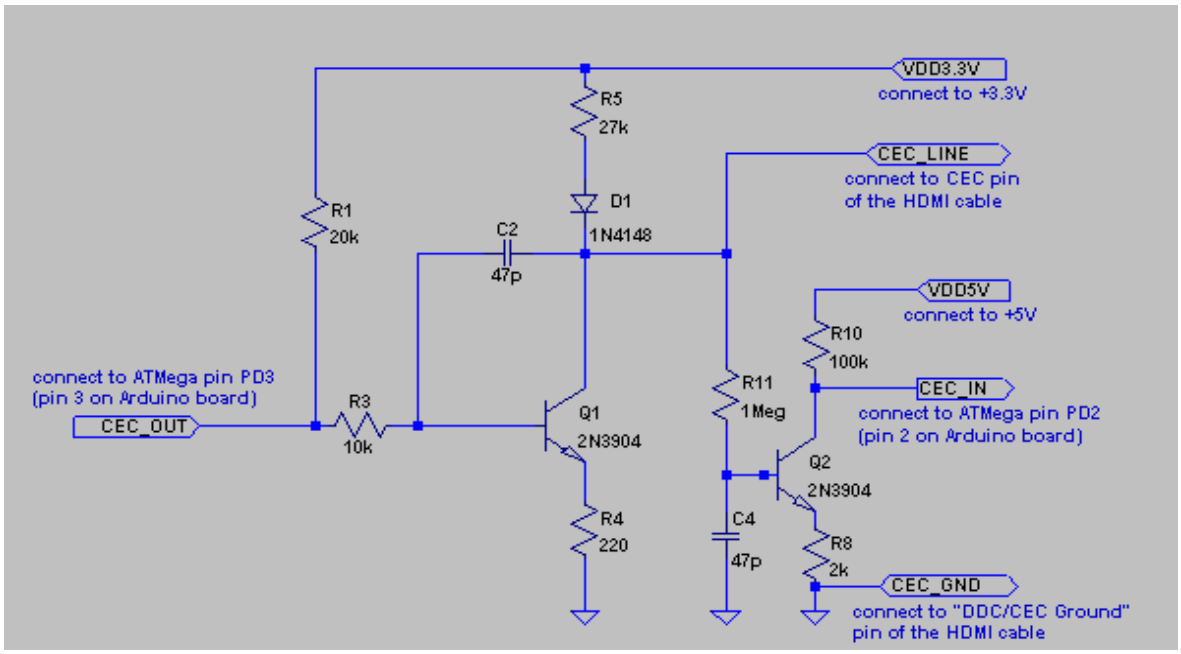

Figure 5: Arduino CEC interface

Figure 6: USB-CEC Adapter from Pulse Eight

Figure 7: USB-CEC Bridge from RainShadow Tech

The pulse Eight device has an accompanying C library called libCEC[15], however, the RainShadow Tech device provides a simple serial interface that can more easily be controlled using Python, which was the language of choice to develop the fuzzer – CECSTeR.

5.3 The CECSTeR tool

CECSTeR is an acronym for **C**onsumer **E**lectronics **C**ontrol **S**ecurity **Te**sting **R**esource. It is a Python-based test tool for sending arbitrary CEC and CDC (see section 5.1) commands, capturing CEC network traffic and fuzzing the CEC and CDC protocols.

Installation

CECSTeR can be downloaded from http://tinyurl.com/ncctools

There are a number of prerequisites that need to be downloaded and installed in order to use CECSTeR; these are:

- Python - http://www.python.org/getit/
- wxPython - http://www.wxpython.org/download.php#stable

Usage

To launch CECSTeR you need to provide the COM port for the RainShadow Tech USB-CEC bridge, as shown in Figure 8.

```
C:\>CECSTeR.py

-----------------------------------------------
Usage: CECSTeR <COM port>
-----------------------------------------------
```

Figure 8: CECSTeR usage

The example use-cases of CECSTeR that follow are based on the CEC network in Figure 9.

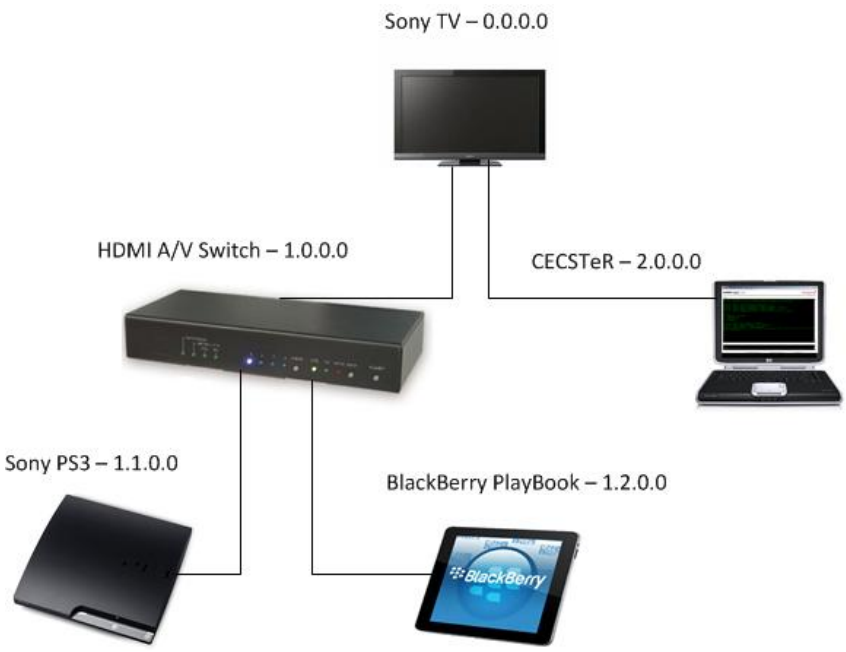

Figure 9: Test CEC network used for demonstrating CECSTeR

When CECSTeR is first started up, the user interface, as shown in Figure 10 is displayed. As can be seen, the default logical address for CECSTeR is 4 (Playback 1) and the default target logical address is 0 (TV).

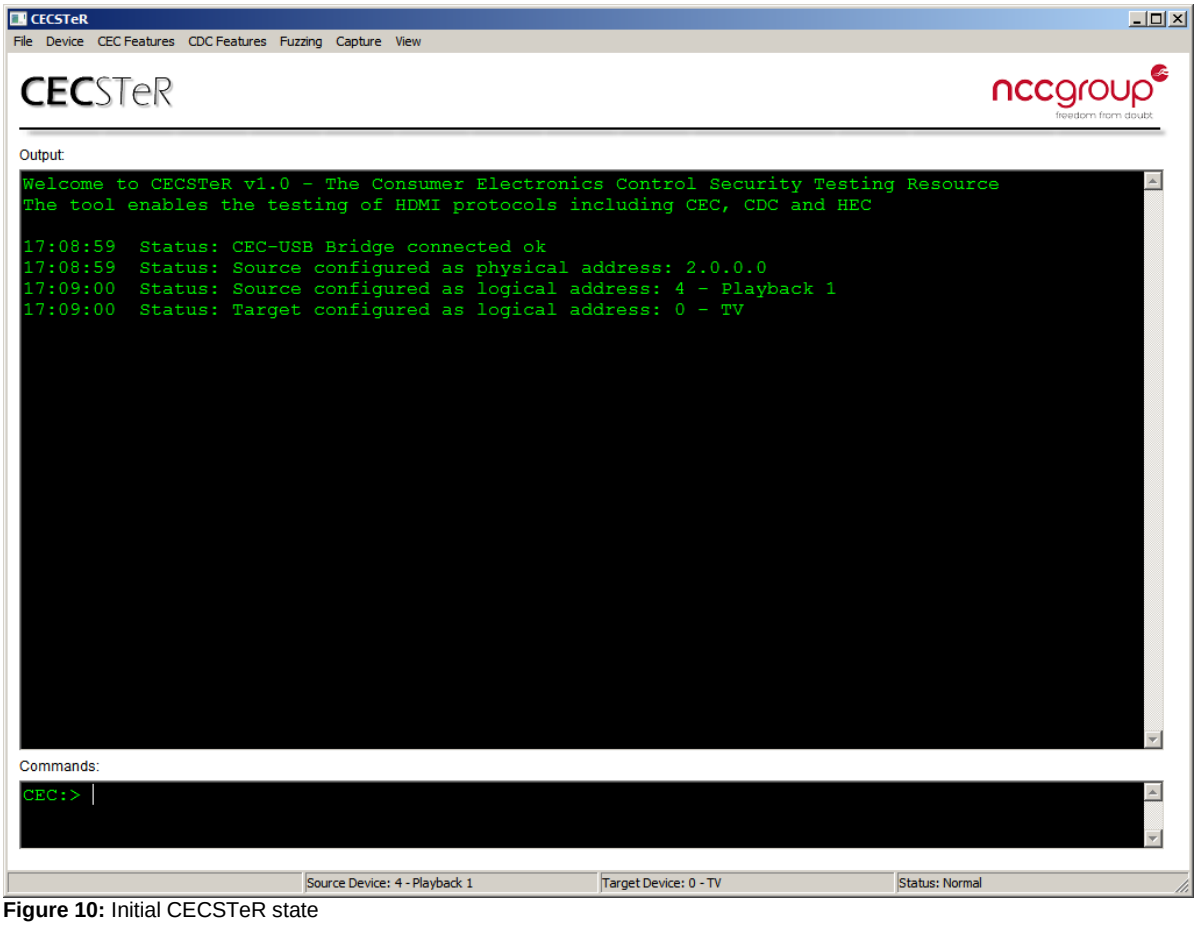

Figure 10: Initial CECSTeR state

The first procedure you would normally want to perform is “Device” > “Enumerate Devices”, which will enumerate the CEC bus and display all connected devices. This is shown in Figures 11 and 12.

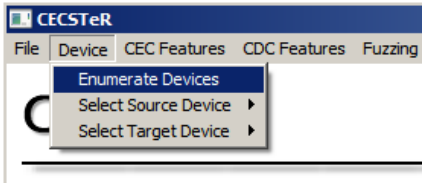

Figure 11: Identifying what is connected in the CEC network

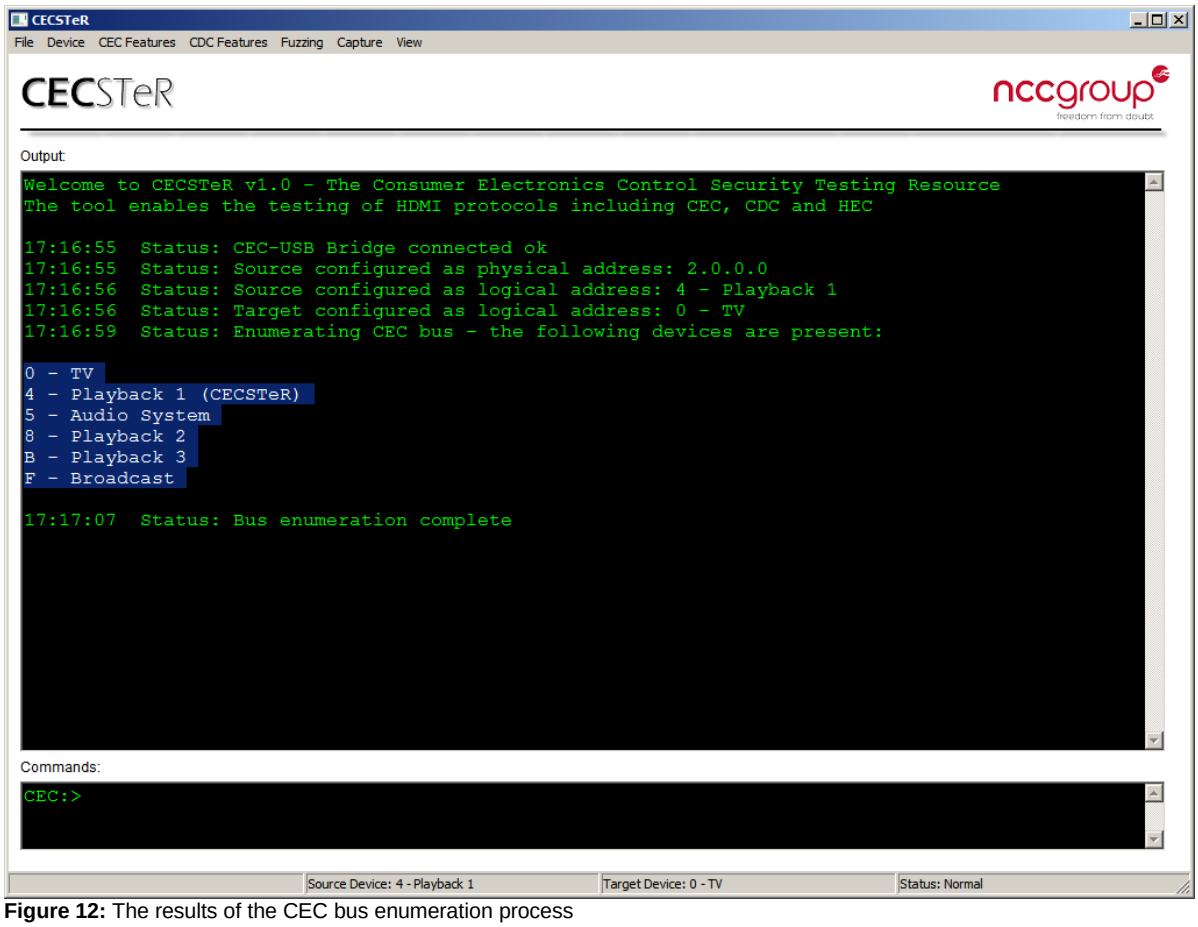

Figure 12: The results of the CEC bus enumeration process

Now that the CEC bus has been enumerated, the target device can be selected using “Device” > “Select Target Device”, as shown in Figure 13.

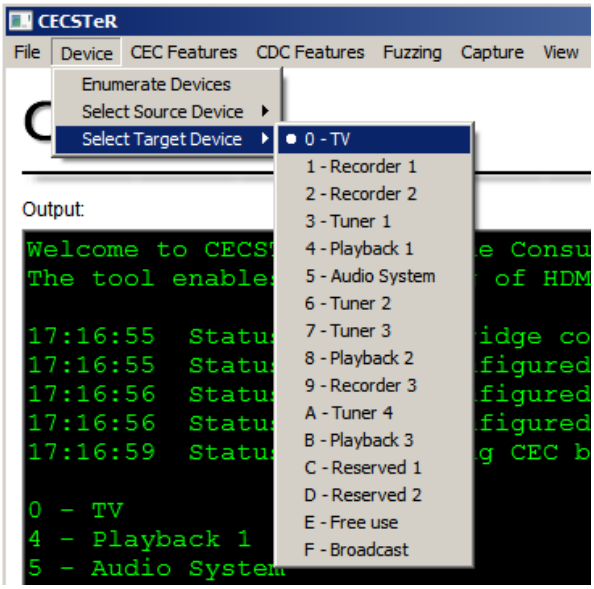

Figure 13: Selecting a new target device

With the target device selected, arbitrary CEC (or CDC) Features can now be executed. The example shown in Figures 14 and 16 show the “GET CEC Version” Feature being executed by selecting “CEC Features” > “Vendor Specific Commands”.

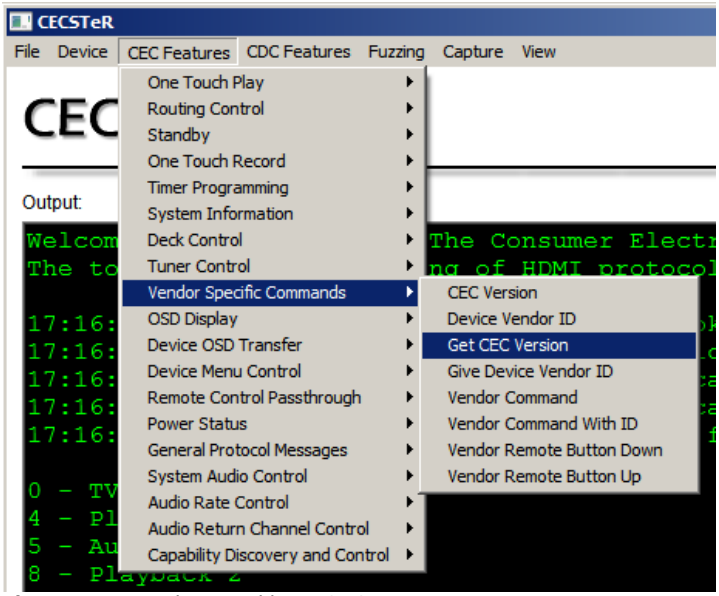

Figure 14: Executing an arbitrary CEC Feature

When a feature has been selected, it is shown in the "Command" window (Figure 15). If operand data needs to be supplied it can be entered here before pressing <return> otherwise, just pressing <return> will execute the Feature. Rather than using the menu, another approach is to directly type the opcode followed by any operand data into the "Command" window and press <return> to execute.

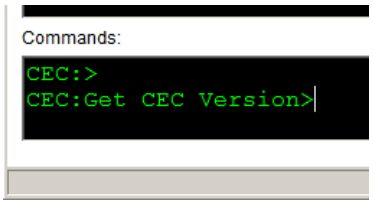

Figure 15: Using the command window

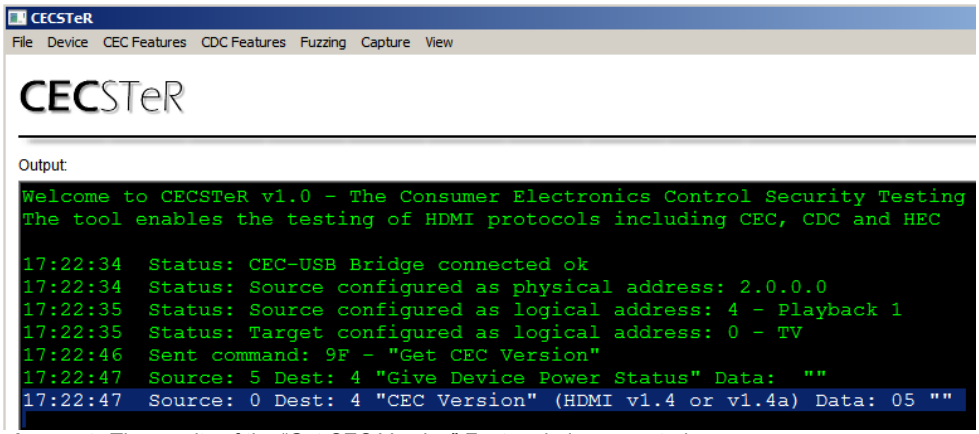

Figure 16: The results of the "Get CEC Version" Feature being executed

The entire CEC Feature set for the selected target can be fuzzed. The fuzzer is currently relatively basic in nature – each CEC Feature is fuzzed consecutively with whatever mutator class has been selected. The current mutator classes are:

- Long strings
- Format strings
- Bit flipping

Also, currently there is no instrumentation implemented, however this will most likely be added in a future version of CECSTeR.

To start the fuzzer, first, select the mutators from “Fuzzing” > “Fuzzing Config” (Figure 17) and then select “Fuzz CEC”. The results are shown in Figure 18.

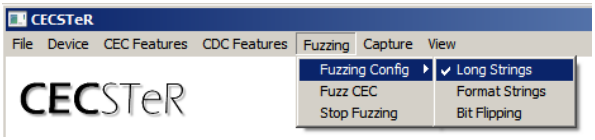

Figure 17: Configuring the fuzzer

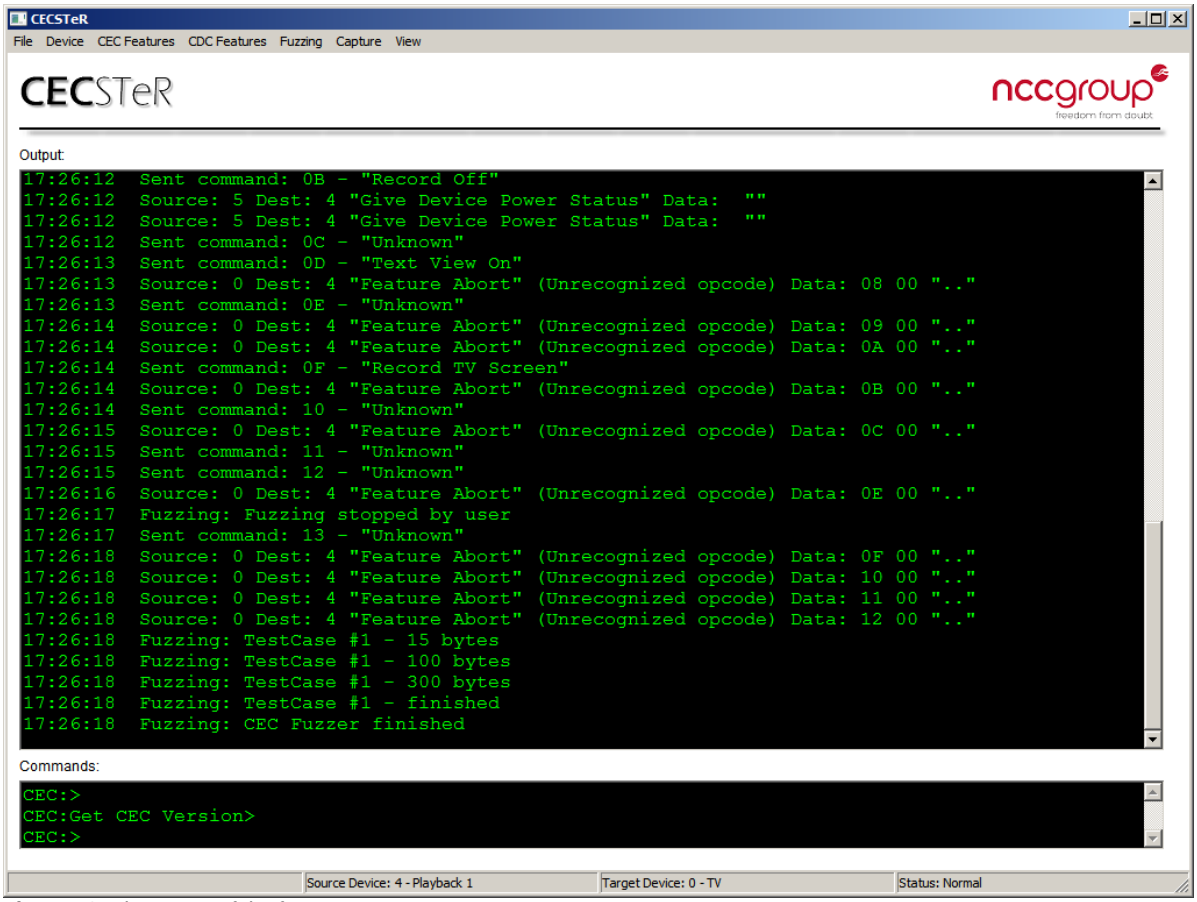

Figure 18: The output of the fuzzer

As all CEC bus traffic is visible to all connected devices, CECSTeR also includes a network capture feature, which is started with “Capture” > “Start Capture” (Figure 19). The output of the capture is displayed in real-time and an example is shown in Figure 20.

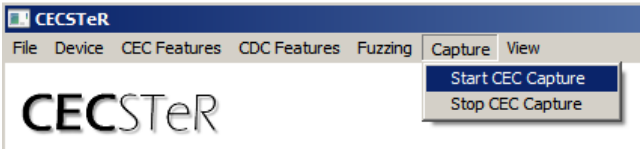

Figure 19: Starting a network capture

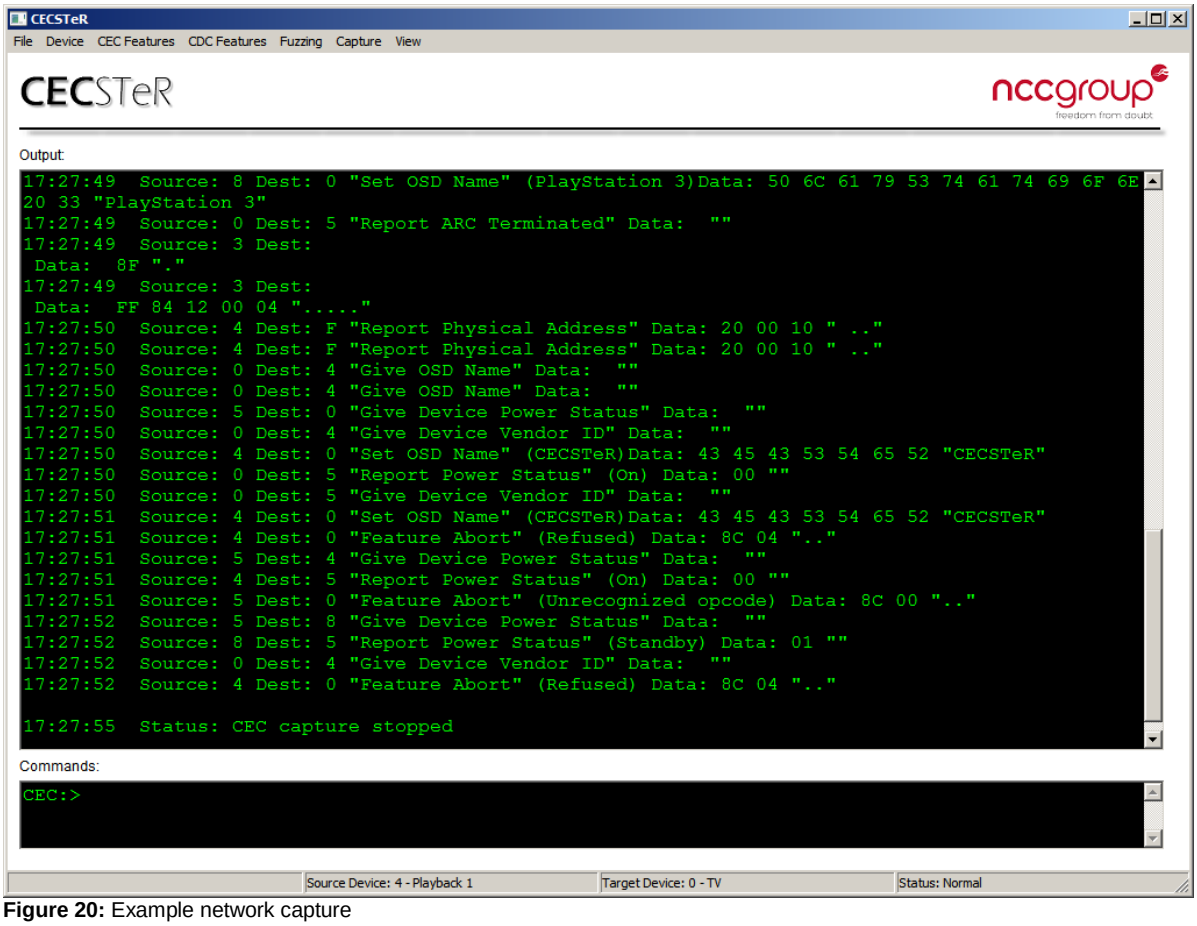

Figure 20: Example network capture

Finally, there are some options for how the data is displayed, the “View” menu allows the user to select if “Polling Messages” (this is a simple mechanism that is used to determine if devices are connected to a CEC network, however, unlike the ICMP[16] protocol, there is no scope for any information leakage bugs) or “Raw Data” are displayed or suppressed (Figure 21).

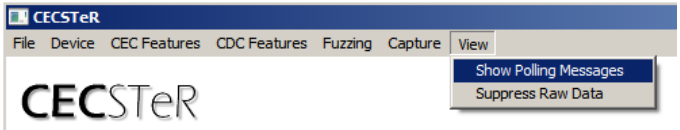

Figure 21: Selecting View options

5.4 Initial fuzzing results

The fuzzing attempts undertaken by NCC Group have so far have been limited to a small number of CEC-aware devices; these are:

- Sony PS3
- Panasonic Blu-ray player (DMP-BD45)
- BlackBerry PlayBook
- XBMC (using Pulse-eight USB-CEC Adapter)

The Sony PS3 and BlackBerry PlayBook both yielded no results, but both devices only implement a very small subset of the CEC Feature set. Fuzzing the Panasonic Blu-ray player resulted in a number of “random” lockups, but when XBMC was tested, the fuzzing resulted in a permanent Denial of Service to the Pulse-eight adapter (the device could not be recovered, even after a power cycle), after one of the buffer overflow test cases was sent.

There is clearly benefit from more being be done against CEC-aware devices, but even with the limited results presented here, it shows that bugs can be triggered as a result of malformed CEC data. If an exploitable memory corruption bug were to be found in a video driver it is likely that the code would be running with high privileges, as they often run with kernel privileges. There is also the possibility that CEC vulnerabilities may be discovered in user-land applications that process CEC data.

6 HDMI Ethernet Channel

HDMI Ethernet Channel was introduced in HDMI v1.4, it consolidates video, audio, and data streams into a single HDMI cable. The Ethernet channel is physically combined with audio return functionality onto a single pair of wires (or pins HEAC+ and HEAC-) on the HDMI connector. This single pair of wires can carry Ethernet only, audio only, or both simultaneously. The primary intention of HEC is to reduce the amount of cables required to connect A/V devices together.

6.1 Capability Discovery and Control

HEC uses the Capability Discovery Control protocol to control Ethernet channels. More specifically, CDC is used to:

- Discover potential HDMI Ethernet channels
- Activate and deactivate channels
- Communicate the status of channels

CDC messages are sent within CEC messages, using the “CDC Message” (0xF8) opcode, all messages are sent to the CEC logical broadcast address (0xF) and the CEC physical address of the initiator is also included prior to the opcode and operand. The structure of a CDC message when encapsulated in a CEC message is shown in Figure 22.

Figure 22: CEC/CDC message

There are seven CDC message types within the HDMI Specification v1.4a, some of which require operand data and some of which do not.

6.2 HEC (CDC) messages

The following messages are used for Capability Discovery and Control:

- `<CDC_HEC_InquireState>` - What's the state of an Ethernet channel?
- `<CDC_HEC_ReportState>` - The state of an Ethernet channel is…
- `<CDC_HEC_SetState>` - Set the state of an Ethernet channel
- `<CDC_HEC_RequestDeactivation>` - Deactivate and Ethernet channel
- `<CDC_HEC_NotifyAlive>` - Keep "active" Ethernet channels active
- `<CDC_HEC_Discover>` - Discover all HEC capabilities of devices within the network
- `<CDC_HEC_SetStateAdjacent>` - Activate or deactivate an Ethernet channel in an adjacent device

6.3 HEC states

There are three different states an Ethernet channel can be in; these are PHEC, VHEC and AHEC:

- PHEC (Potential HDMI Ethernet Channel) – the device is part of a PHEC if at least one HDMI connection is HEC capable
- VHEC (Verified HDMI Ethernet Channel) – the device is part of a VHEC after CDC has confirmed HEC capability of all devices in a PHEC via a `<CDC_HEC_Discover>` message
- AHEC (Active HDMI Ethernet Channel) – the device is part of an AHEC after activation of all devices in a VHEC via a `<CDC_HEC_SetState>` message.

6.4 Possible HECs within a certain HDMI network

Within a given HDMI network there are a large number of different permutations of possible Ethernet channel networks. Figure 23 shows some examples of different Ethernet channel network combinations for a single HDMI network architecture.

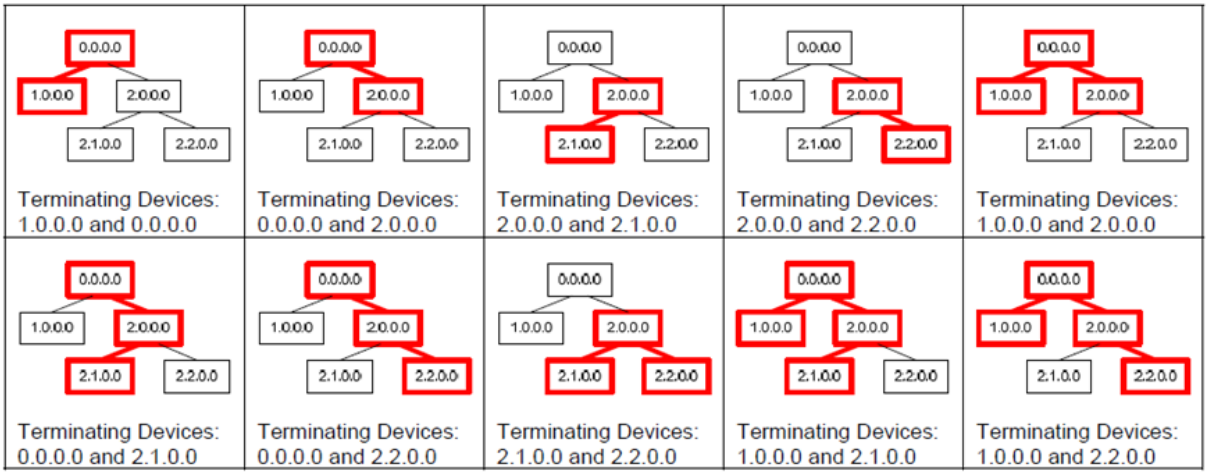

Figure 23: Different combinations of possible Ethernet Channel networks

6.5 Network loop prevention

By allowing such flexibility with regard to how HDMI Ethernet networks are connected, it is easy to encounter routing loops, which negatively impact on the performance of the network. Routing loops such as shown in Figures 24, 25 and 26 are managed using RSTP (Rapid Spanning Tree Protocol).

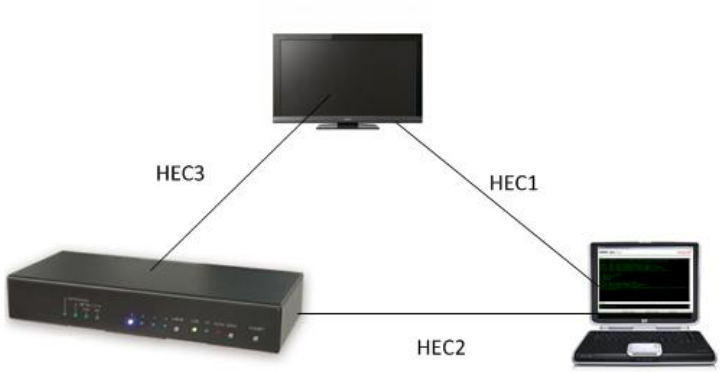

Figure 24: An HDMI Ethernet network loop

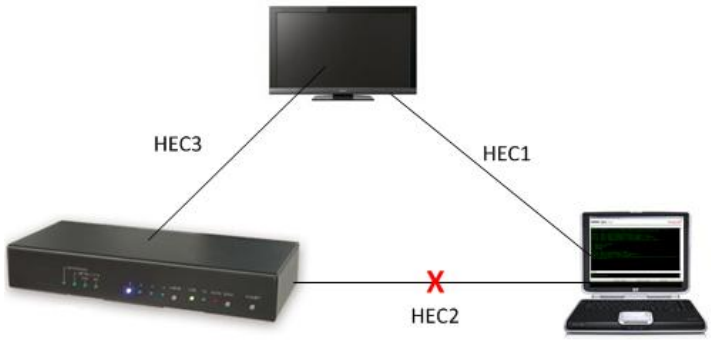

Figure 25: HEC2 is disabled to remove the loop

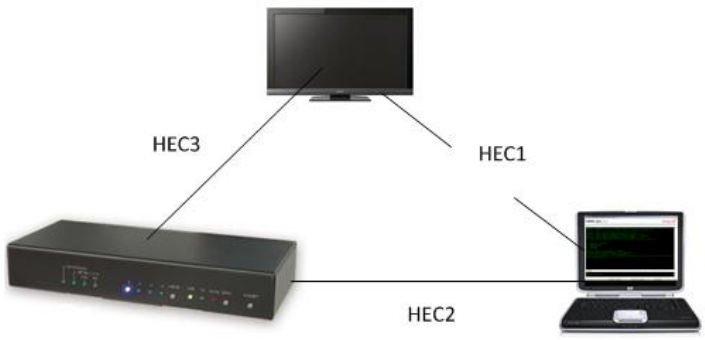

Figure 26: If HEC1 link is broken, HEC2 is restored

6.6 Queue control

Devices in a HEC network are expected to prioritise traffic. Time sensitive application traffic should be forwarded with higher priority than activities such as file downloads e.g.:

- On-line gaming
- Video
- Voice-over-IP

This is achieved using a 3 bit priority field in VLAN tags within the layer 2 network frames, as shown in Figure 27. The HEC user priorities are shown in Table 2.

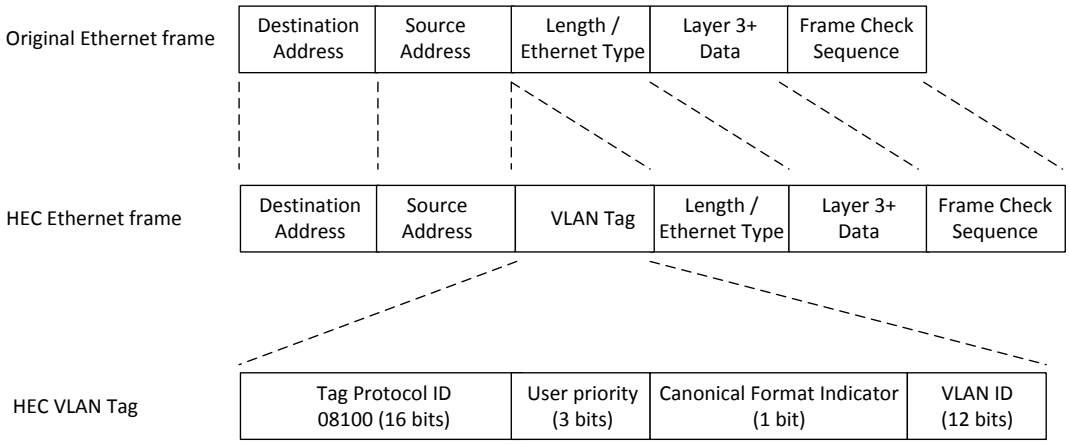

Figure 27: HEC VLAN Tags

Traffic type or Application type	User priority
Control	7
A/V streaming	5
Data services	2
Best effort	0

Table 2: HEC user priorities

6.7 HDMI corporate risks

At this point the reader may be thinking “This is all very interesting, but I’m never going to be security testing a home A/V network” or “If I found a bug in an HDMI enabled TV, so what?” These are fair points, however, the only reason this research has focussed on consumer A/V equipment is that this is where the technology is currently being used (it has not yet been implemented in any of the major operating systems). HDMI connectors are appearing on new laptops and PCs so soon these protocols will be implemented by the OS vendors and therefore, these technologies will eventually be deployed in a corporate environment. Plasma/LCD TVs are already becoming part of the corporate network infrastructure, as most modern TVs include Ethernet connectivity to enable software updates and more interactive online content. So how could HDMI protocols affect corporate users?

Corporate Risk #1 – Unauthorised connectivity to a corporate network

Often in a corporate environment visitors will bring their own laptops to meeting rooms to deliver presentations on either a projector or a large-screen TV. Both of these display technologies are commonly network-enabled and often connected to the corporate network. If HEC capabilities are supported by the display device and the visitor's laptop there is a distinct possibility that the visitor's laptop may become connected to the corporate network where it could introduce (inadvertently or otherwise) malware onto the network. This scenario is shown in Figure 28.

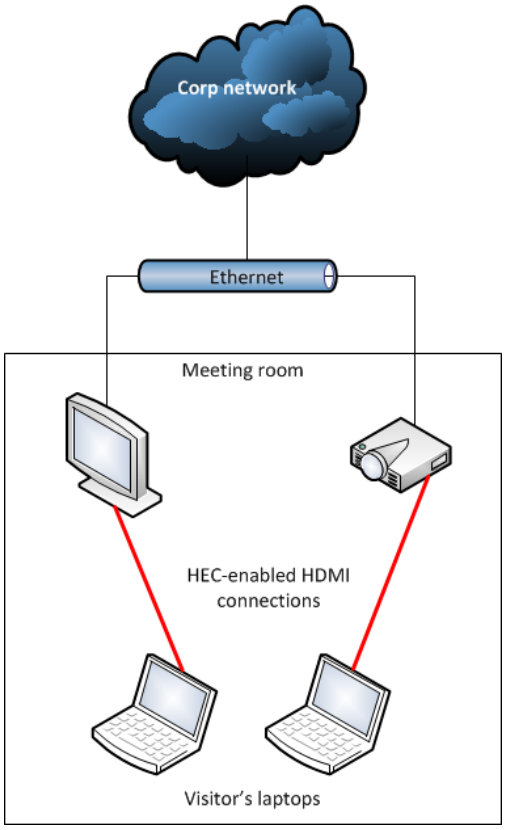

Figure 28: Gaining access to a corporate network via HDMI

It is recommended that HDMI Ethernet channel functionality is disabled in all corporate display devices and that any an Ethernet-connected devices are connected within a "display devices" DMZ rather than directly connected to a corporate network.

HDMI could be used to connect unauthorised network-enabled devices to corporate laptops. If these devices and the corporate laptops support HDMI Ethernet Channel then the devices could be used to introduce malware or exfiltrate sensitive data, as shown in Figure 29.

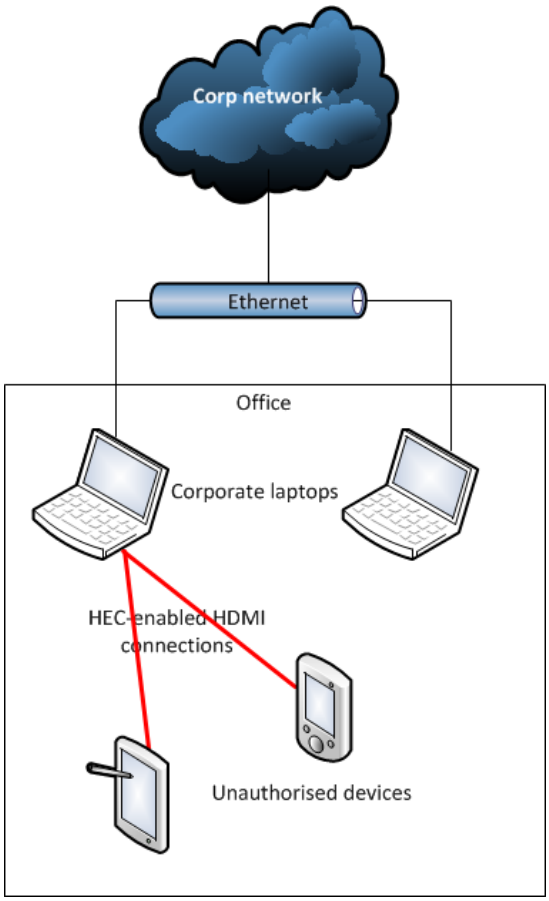

Figure 29: Unauthorised devices connected to a corporate laptop via HDMI

It is recommended that HDMI Ethernet Channel functionality is disabled in all corporate laptops (when HEC has been adopted by the major OS vendors) and if Endpoint Protection software is in use (and HEC functionality has been adopted), it should be configured to prevent devices from using HEC to connect to the host.

Corporate Risk #3 – Unauthorised network extension

HDMI could be used to create an unauthorised extension to the corporate network. The corporate laptop would effectively act as a NAT gateway to the network or unauthorised machines. The machines in this “private network” would not be visible to corporate network monitoring tools / NIDS (Network Intrusion Detection System) devices. This scenario is shown in Figure 30.

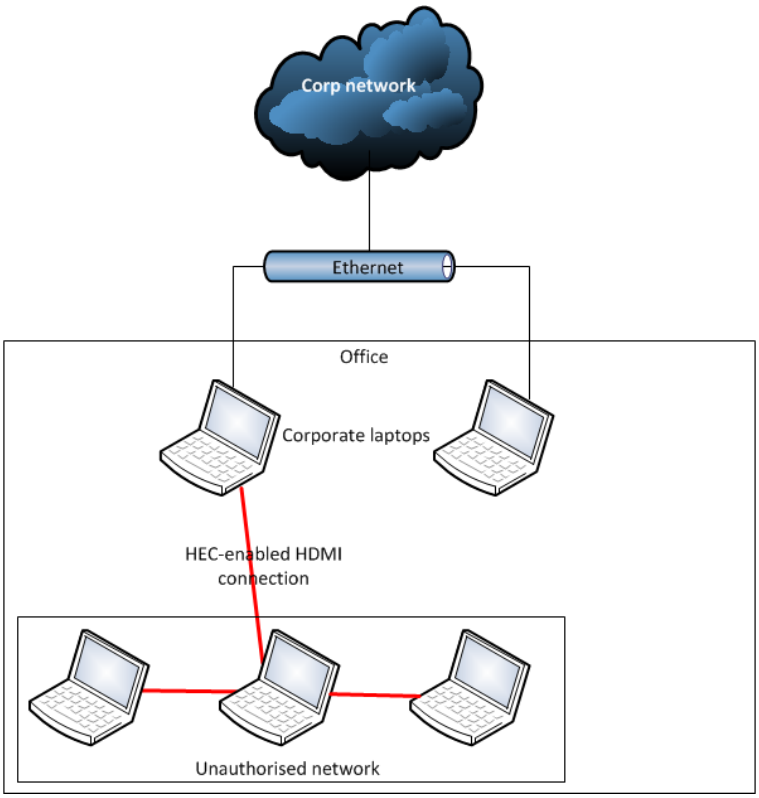

Figure 30: Unauthorised corporate network extension via HDMI

As with risk #2, it is recommended that HDMI Ethernet Channel functionality is disabled in all corporate laptops and if Endpoint Protection software is in use, it should be configured to prevent devices from using HEC to connect to the host.

7 Conclusion

As users demand more and more “seamless” functionality in a plug-and-play world there will be a greater need for bi-directional data to be flowing in A/V links between devices. Technologies such as CEC and HEC will be adopted by the major operating systems to enable the use of HTPCs (Home Theatre PCs) and other A/V functionality.

The Consumer Electronics Control protocol could potentially be abused to exploit memory corruption bugs in software if these are discovered using tools such as CECSTeR. If successfully exploited these bugs may provide remote or elevated access to devices via HDMI. However, most mainstream OS vendors have over the last few years started to implement exploit mitigation strategies, such as DEP (Data Execution Prevention) and ASLR (Address Space Layout Randomisation) to raise the bar of capability required to successfully exploit software vulnerable to memory corruption bugs. Information leakage bugs, which used to be considered much lower impact, are now required in order to defeat ASLR. Furthermore, if identified bugs reside in video software another attack scenario would be gaining access current or previous video content.

Furthermore, HDMI Ethernet Channel could circumvent corporate security in a number of ways, such as through the use of network-enabled display technologies and via HEC-enabled corporate laptops, but the technology is still largely unsupported.

Unfortunately, during the course of this research it has not been possible to actually test HDMI Ethernet Channel in any “real world” scenario, as the protocol is so new it has only been adopted by a few extremely high-end devices, costing thousands of pounds e.g. The T+A Blu-ray receiver[17]. However, this situation is likely to change over the next few years.

8 CEC Features

This section provides a list of the currently supported CEC Features and their associated opcodes.

"Active Source" - 0x82
"Image View On" - 0x04
"Text View On" - 0x0D
"Inactive Source" - 0x9D
"Request Active Source" - 0x85
"Routing Change" - 0x80
"Routing Information" - 0x81
"Set Stream Path" - 0x86
"Standby" - 0x36
"Record Off" - 0x0B
"Record On" - 0x09
"Record Status" - 0x0A
"Record TV Screen" - 0x0F
"Clear Analogue Timer" - 0x33
"Clear Digital Timer" - 0x99
"Clear External Timer" - 0xA1
"Set Analogue Timer" - 0x34
"Set Digital Timer" - 0x97
"Set External Timer" - 0xA2
"Set Timer Programme Title" - 0x67
"Timer Cleared Status" - 0x43
"Timer Status" - 0x35
"CEC Version" - 0x9E
"Get CEC Version" - 0x9F
"Give Physical Address" - 0x83
"Get Menu Language" - 0x91
"Report Physical Address" - 0x84
"Set Menu Language" - 0x32
"Deck Control" - 0x42
"Deck Status" - 0x1B
"Give Deck Status" - 0x1A
"Play" - 0x41
"Give Tuner Device Status" - 0x08
"Select Analogue Service" - 0x92
"Select Digital Service" - 0x93
"Tuner Device Status" - 0x07
"Tuner Step Decrement" - 0x06
"Tuner Step Increment" - 0x05
"Device Vendor ID" - 0x87
"Give Device Vendor ID" - 0x8C
"Vendor Command" - 0x89
"Vendor Command With ID" - 0xA0
"Vendor Remote Button Down" - 0x8A
"Vendor Remote Button Up" - 0x8B
"Set OSD String" - 0x64
"Give OSD Name" - 0x46

```
"Set OSD Name" - 0x47
"Menu Request" - 0x8D
"Menu Status" - 0x8E
"User Control Pressed" - 0x44
"User Control Released" - 0x45
"Give Device Power Status" - 0x8F
"Report Power Status" - 0x90
"Feature Abort" - 0x00
"Abort" - 0xFF
"Give Audio Status" - 0x71
"Give System Audio Mode Status" - 0x7E
"Report Audio Status" - 0x7A
"Report Short Audio Descriptor" - 0xA3
"Request Short Audio Descriptor" - 0xA4
"Set System Audio Mode" - 0x72
"System Audio Mode Request" - 0x70
"System Audio Mode Status" - 0x7E
"Set Audio Rate" - 0x9A
"Initiate ARC" - 0xC0
"Report ARC Initiated" - 0xC1
"Report ARC Terminated" - 0xC2
"Request ARC Initiation" - 0xC3
"Request ARC Termination" - 0xC4
"Terminate ARC" - 0xC5
"CDC Message" - 0xF8
```

9 CDC Messages

This section provides a list of the currently supported CDC messages and their associated opcodes.

```
"CDC_HEC_InquireState" - 0x00
"CDC_HEC_ReportState" - 0x01
"CDC_HEC_SetState" - 0x02
"CDC_HEC_RequestDeactivation" - 0x03
"CDC_HEC_NotifyAlive" - 0x04
"CDC_HEC_Discover" - 0x05
"CDC_HEC_SetStateAdjacent" - 0x06
```


10 References & further reading

1 - http://www.bunniestudios.com/blog/?p=2117
2 - http://www.nccgroup.com/en/learning-research-centre/security-testing-audit-compliance-resources/white-papers/
3 - http://en.wikipedia.org/wiki/Video_Graphics_Array
4 - http://en.wikipedia.org/wiki/Extended_display_identification_data
5 - http://xtreamerdev.googlecode.com/files/CEC_Specs.pdf
6 - http://www.hdmi.org/manufacturer/hdmi_1_4/hec.aspx
7 - http://www.digital-cp.com/
8 - http://www.hdmi.org/manufacturer/specification.aspx
9 - http://en.wikipedia.org/wiki/Transition-minimized_differential_signaling
10 - http://en.wikipedia.org/wiki/Display_Data_Channel
11 - http://en.wikipedia.org/wiki/AV.link
12 - http://www.quantumdata.com/pdf/EDID.pps
13 - http://www.pulse-eight.com/store/products/104-usb-hdmi-cec-adapter.aspx
14 - http://rainshadowtech.com/downloads/HDMICECtoUSBandRS232v7.pdf
15 - http://libcec.pulse-eight.com/
16 - http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
17 - http://studio-hifi.com/images/K8reviewHoneCinemaChoice04.2012.pdf

11 Glossary

A/V – Audio/Visual
AHEC – Active HDMI Ethernet Channel
CDC – Capability Discovery and Control
CEC – Consumer Electronics Control
DDC – Display Data Channel
DMZ – De-Militarised Zone
EDID – Extended Display Identification Data
HDCP – High-bandwidth Digital Content Protection
HDMI – High Definition Multimedia Interface
HEC – HDMI Ethernet Channel
HTPC – Home Theatre Personal Computer
ICMP – Internet Control Message Protocol
LCD – Liquid Crystal Display
NAT – Network Address Translation
NIDS – Network Intrusion Detection System
PHEC – Potential HDMI Ethernet Channel
RSTP – Rapid Spanning Tree Protocol
TMDS – Transition-Minimized Differential Signalling
VHEC – Verified HDMI Ethernet Channel
VLAN – Virtual Local Area Network
VGA – Video Graphics Array

