

WHITE PAPER

USING A KVM TO HACK INTO A SECURE ENVIRONMENT

HighSecLabs
Highest Security Solutions

HLT33874 Rev 1.0

TABLE OF CONTACTS

1. Introduction 3
1.1 Overview 3
1.2 Document Scope 4
1.3 Environment Description 4

2. Attack Overview 6
2.1 Attack Illustration 6
2.2 Attack Objectives 7
2.3 Attack Logic and Claims 7
2.4 Attack Plan 10

3. The Security Hole 11
3.1 How are Monitor Settings Saved? 12
3.2 Ways to Exploit Monitor Controls 12
3.3 Steps to bidirectional computer-to-monitor exploitation 16

4. Step-by-Step Workflow 17
4.1 Workflow Example 1: Exploit Valid Values 17
4.2 Workflow Example 2: Manipulate Valid Controls to Save Non-Valid Values 21

5. How HSL KVMs Secure the Digital Video Communication 27
5.1 Overview 27

1. INTRODUCTION

1.1 OVERVIEW

Security-cognizant organizations undertake efforts to protect valuable assets from the reach of the outside world. Protecting computer environments and digital assets such as usernames, passwords, documents, pictures, audio, video, and other data is especially significant.

Physically separating classified and non-classified computer environments is a common practice to achieve complete isolation and prevent data leakage. While physical segregation is believed to effectively provide high security, it interferes with the end users' work, requiring simultaneous access to information on both the secure and the non-secure environments.

A KVM switch is a convenient tool for the end user to seamlessly switch between computers operating in different security environments. At the same time, however, the KVM itself and the keyboard, video, and mouse peripherals common to all environments are all security vulnerabilities.

Peripherals and KVMs are vulnerable to security attacks, and can therefore become a security breach hazard. Once shared between multiple computers, a single peripheral vulnerability is enough to cause data leakage between different security environments. The KVM can cause a security breach when it is vulnerable to firmware re-programing, keyboard strike logging, and other intrusion methods.

Since KVMs and their related peripherals are a security hazards, there is a pressing need for Secure KVMs as an alternative to non-secure commercial KVMs. A Secure KVM includes both hardware and software features that completely segregate all computers connected through it, preventing the KVM and peripherals from becoming points of failure.

1.2 DOCUMENT SCOPE

This document demonstrates the real danger of KVMs and peripherals becoming security hazards. Even without physical access to the KVM, the peripherals, or to any of the connected computers, an intruder can exploit many of the vulnerabilities of a commercial KVM and the connected peripherals to 'breach the barrier' and transfer data between secure and non-secure computers.

Part 1 of this document includes a detailed step-by-step workflow. It describes how to exploit a standard Dell monitor by saving stolen data into the monitor's internal memory and sharing it between two host computers.

Part 2 of this document shows how the HSL Secure KVM eliminates such threats.

Notes:

- Key steps were purposely left out of the procedures described in this document to prevent improper use of the information.
- The use of a Dell monitor in this document does not imply in any way that it has special characteristics or weaknesses beyond those of any other common monitor brands available today. It was chosen due to the popularity of Dell products in business environments.

1.3 ENVIRONMENT DESCRIPTION

This exercise simulates a common work environment with two end user PCs.

Each PC is connected to a completely independent network infrastructure of switches, servers, power grid, and other resources. A commercial 2-port KVM switch connects both computers to the following peripherals:

- Generic USB English Keyboard (K)
- Dell U2412mb Monitor (V)
- Generic USB Mouse (M)

PC#1 is considered secure. It is connected to a '**Red**' maximum security environment, having no access to external networks. This computer is dedicated to confidential tasks, such as working on classified documents, accessing classified databases, and internal applications.

PC#2 is not secure. It is connected to a '**Green**' minimum security environment and has internet access. Users are authorized to use this computer for web browsing and communication with the outside world via email, instant messaging, VoIP (Skype, video chat, and others), and file sharing services (e.g., Google Drive, Dropbox, OneDrive). Any access or use of confidential documents, videos, applications, and other information is strictly prohibited in a "Green" environment.

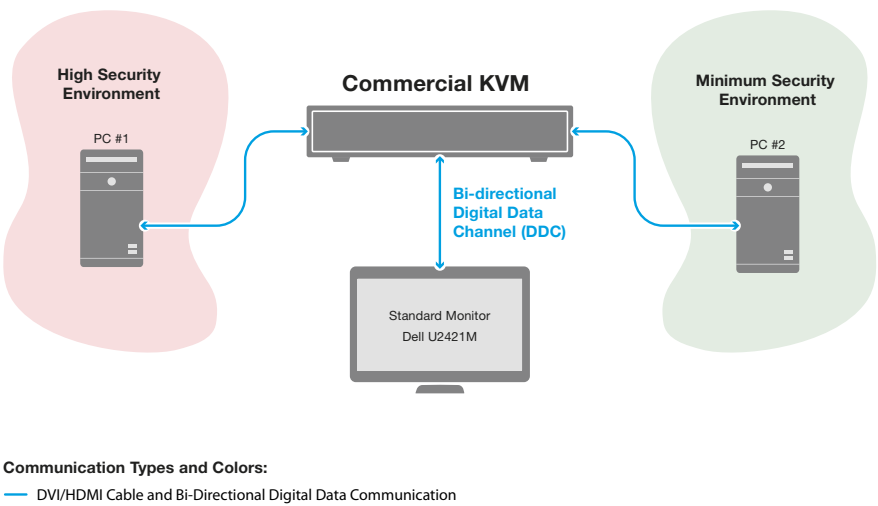


***Figure 1**. Secure and Non-Secure Computers in a Common Environment*

2. ATTACK OVERVIEW

The following sections describe the attack:

- Objectives
- Logic
- Arguments
- Plan

2.1 ATTACK ILLUSTRATION

Figure 2 below shows how data is transferred through the monitor between two computers that are both connected to a commercial KVM.

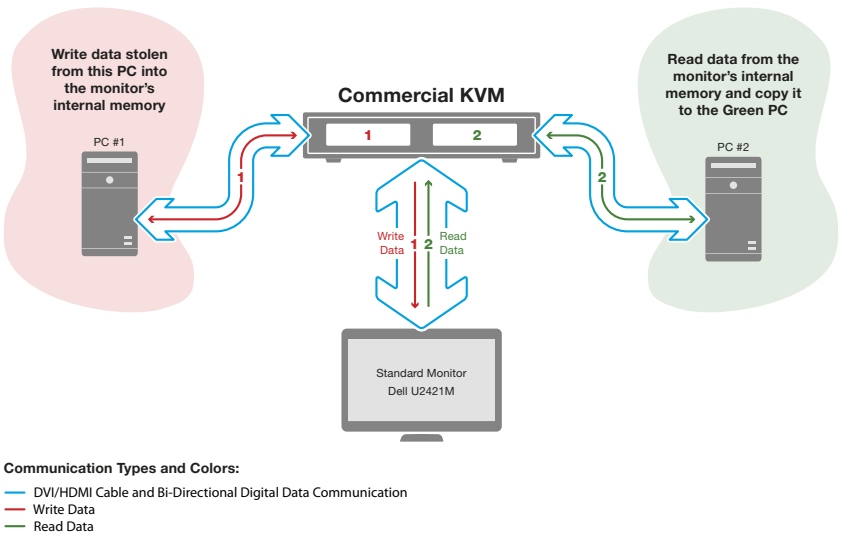


***Figure 2**. General Illustration of Attack*

2.2 ATTACK OBJECTIVES

Table 1 below describes the attack objectives.

#	Objective	Description
1	Stealth	Attack must be undetectable
2	Breach segregation within the physical environment	Expose Red and Green environments to each other
3	Transfer data	Transfer data between computers in both directions

***Table 1**. Attack Objectives and Descriptions*

2.3 ATTACK LOGIC AND CLAIMS

"Shared peripherals are vulnerable to security attacks."

2.3.1 BASE ASSUMPTION:

A standard monitor shared between two computers via a KVM is a security vulnerability. Why?

- Two computers share the same monitor.
- Each computer, in turn, can read and write data into the monitor's memory.
- The computer-to-monitor communication is done via low-level functions transmitted undetectably over the monitor's DVI/HDMI cable which is unmonitored by antivirus and intrusion detection and prevention systems (IDPS).
- Antivirus and IDPS systems check suspicious files and activities against predefined databases. Peripheral and KVM malicious code is likely to remain untraceable, as it is tailored uniquely to a target peripheral and therefore would not fit with common AV/IDPS database definitions.

2.3.2 HOW?

A monitor with a digital video interface (DVI and/or HDMI) is seamlessly controlled by the computer connected to it using a standard set of commands. Normally used to adjust common monitor settings such as luminance, contrast, and other settings, the computer sends these commands to the monitor over the DVI/HDMI cable and through the KVM, which connects the computers to the monitor.

An intruder can exploit this bidirectional computer-to-monitor digital communication channel to attack the connected computers by running a program (virus) that writes stolen data from PC#1 into various fields inside the monitor settings memory, where luminance, contrast, and other display-related settings are saved. The same program, executing on PC#2, then reads the information from the monitor, as seen in Figure 1.

2.3.3 EXTENDED DESCRIPTION:

Most monitors support industry standard DCC/CI (Display Data Channel/ Command Interface) protocols to communicate with the computer's graphic adapter. The computer reads the monitor's EDID (Extended Display Identification Data) information and uses VESA MCCS (Monitor Control Command Set) to adjust monitor parameters.

The monitor's EDID information describes its capabilities, including vendor information, maximum image size, color features, frequency range limits, and more. Modern plug-and-play operating systems and software applications require EDID information to operate normally. This information, which is stored in the monitor's internal memory and is used to configure the monitor to work with the PC, passes to the PC through a Display Data Channel (DDC).

Notes:

Monitors have a broad set of vulnerabilities. For simplicity, this document covers a single basic method to exploit MCCS Commands in order to leak data. The HSL Secure KVM handles additional monitor vulnerabilities which are not covered by this document, including:

- EDID backup / restore and signaling
- Asset tag programing
- Monitor firmware upgrades
- Production test registers
- Power management manipulation

2.4 ATTACK PLAN

Table 2 below summarizes the attack plan, including task, order, and description.

Note: Key steps were purposely left out of the procedures described in this document, to prevent improper use of this information.

#	Task	Description	Detailed in this Doc.
1	Take control of the "Green" PC by exploiting its low security settings and internet connectivity	Inject malicious code by email attachment, picture, application, or other common code infection method.	No (-)
2	Steal data from the "Red" (secure) PC and transmit it back to the green PC	Execute a Trojan Horse or virus that can capture data, sniff passwords, execute commands, and inject data.	No (-)
3	Use the bidirectional computer-to-monitor digital communications to execute the attack	Write the stolen data as various monitor values into the monitor's memory settings, where luminance, contrast, and other display related parameters are usually saved.	Yes (√)
4	Take control of the "Red" PC	*This step was purposely left out.	No (-)
5	Use the "Red" PC to steal and transmit data	Execute a Trojan Horse or virus that can capture data, sniff passwords, execute commands, and inject data.	No (-)
6	Use CLI (Command Line Instructions) and hidden scripts to execute commands and transfer data between the computers	Execute DDC MCCS commands to read and write data to and from the monitor's memory. The monitor serves as a broker to store data passed between the computers that are connected to it.	Yes (√)
7	Covertly payload the stolen data to an external destination	Encapsulate and send the data to the an outside destination.	No (-)

***Table 2**. Attack Plan Tasks*

3. THE SECURITY HOLE

The following points summarize the main causes that make a shared monitor a security vulnerability:

1. **Monitor firmware and hardware are not secure by design**, allowing an attacker to manipulate the firmware and save data into the monitor's internal memory. Hardware components that are supposedly programmed with read-only code (such as the EDID chip) are also easy targets for a successful attack.
2. **Commercial KVMs are not designed to enforce security.** They do NOT provide any segregation between connected computers or shared peripherals.
3. **Read and Write commands can pass through.** Once able to read and write values in the monitor's internal memory, an attacker can transfer data between all the hosts that are connected through the KVM.

3.1 HOW ARE MONITOR SETTINGS SAVED?

Monitor settings are saved as values within predefined controls.

A ***Control*** is a configurable field, such as luminance.

A ***Value*** is the data that defines the settings for that control. A value can include digits and letters in various combinations. For example, the values that set the luminance control are numbers that represent the percentage (%) of luminance the monitor displays within a range of 0 – 100.

A unique name and ID number are assigned to each control. For example, the ID number of the "luminance" control is "0x10"

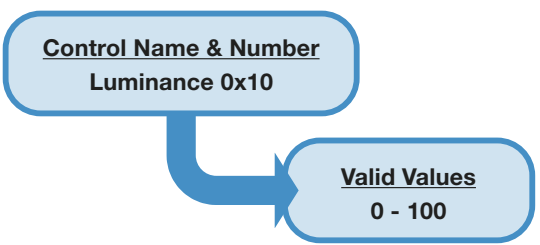


Figure 3. *Control and Value*

3.2 WAYS TO EXPLOIT MONITOR CONTROLS

An attacker can exploit monitor controls in the following ways:

- Exploit valid values to transmit stolen data: For example, change the luminance value from 69% to 70% and then from 70% to 71%, to create a binary stream of 0 and 1 digits.
- Manipulate valid controls to save non-valid values: For example, replace valid control values, such as digits, with non-valid control values, such as letters.

3.2.1 EXAMPLE #1: EXPLOIT VALID VALUES

An easy way to exploit a monitor is by making minor adjustments to valid values in its control settings, such as the luminance values. This method converts data into binary code and then transmits it in binary format (0s and 1s) by making small changes to the selected value settings, such as luminance, contrast, and more.

Changing the luminance settings from 70% to 71% is unnoticeable to the human eye, yet is enough to transmit any amount of binary data, where 70% luminance represents the digit zero (0) and 71% luminance represents the digit one (1). To prevent being detected, these adjustments can be scheduled to execute during off-work or idle hours, when both computers are not in use.

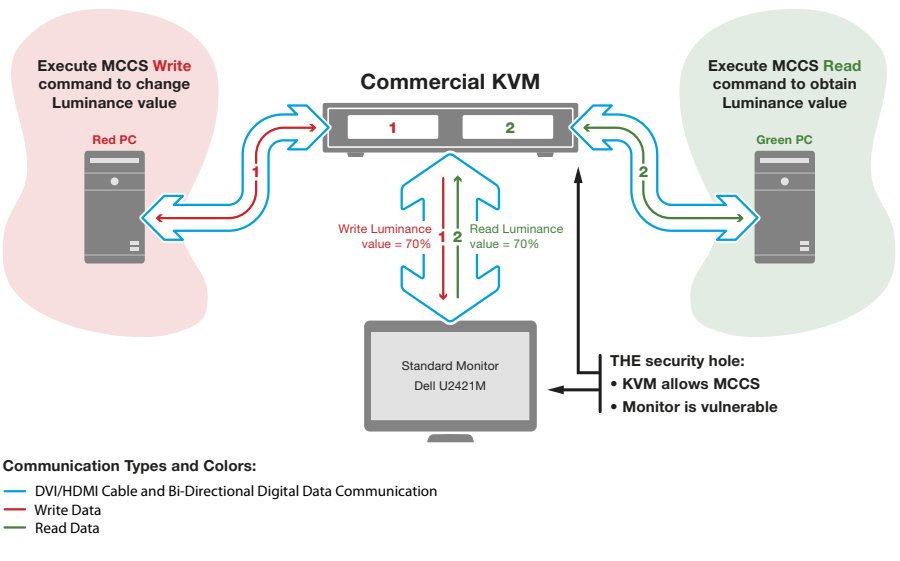


***Figure 4**. Exploit Valid Values*

3.2.2 EXAMPLE #2: MANIPULATE VALID CONTROLS TO SAVE NON-VALID VALUES

The example below shows how stolen data is passed between two computers that share the same monitor via a KVM, by writing and reading false values in a specific control to make unnoticeable changes to the monitor settings. A security vulnerability can be the ability to store non-valid values in the monitor settings, enabling computers to pass information.

The following is required to exploit a monitor:

3.2.2.1 FIND A WRITABLE AND UNNOTICEABLE CONTROL TO ATTACK:

Find a control that does not impact the user experience once changed. 'Last Active Control' 0x52 is such a control, which is used for internal monitor purposes.

3.2.2.2 EXPANDABLE RANGE OF VALUES:

Since standard monitors are not designed with code security for their internal firmware, a control may accept values with no functional purpose. For example, control 0x52 accepts the following values that are NOT part of its standard range of values (F1, F2, F3, F4, F5, F6, F7, F8, F9 and FA, FB, FC, FE), and using them will not impact on the monitor's settings or appearance.

3.2.2.3 RUN A PROGRAM TO READ AND WRITE FALSE VALUES:

Write a program that disassembles the stolen data from one PC into non-valid control values, and write those into the attacked control. On the other PC, read the values from the attacked control and reassemble the data back to its original format. In our example, the non-valid values written into control 0x52 include the digits 1-9 and the letters A-E, which are more than enough to construct words and sentences to steal data. Binary format can be used to transfer data using two digits: zero (0) and one (1).

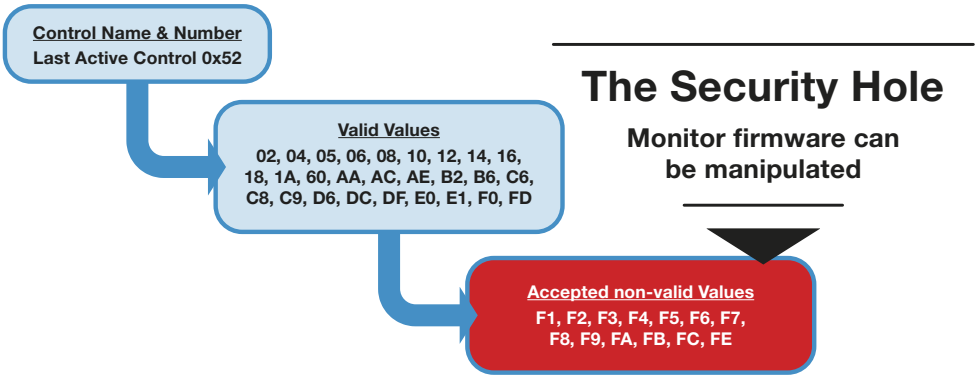


***Figure 5**. Control and Value*

3.3 STEPS TO BIDIRECTIONAL COMPUTER-TO-MONITOR EXPLOITATION

Figure 6 below summarizes the steps on how to exploit a monitor:

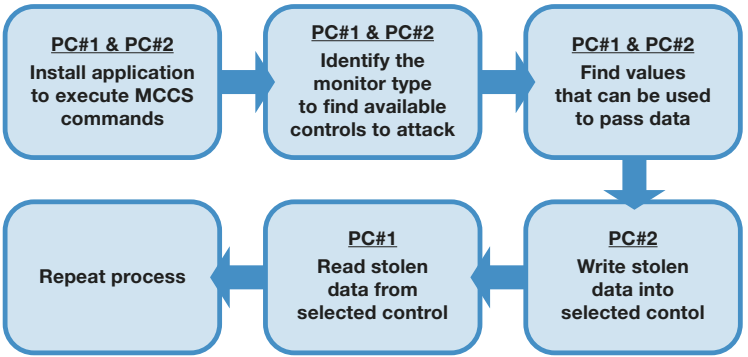


***Figure 6**. Exploitation Process*

4. STEP-BY-STEP WORKFLOW

4.1 WORKFLOW EXAMPLE 1: EXPLOIT VALID VALUES

This procedure shows a step-by-step workflow to exploit the luminance values of a Dell U2412M monitor. The text color represents the computer on which the action takes place, where **RED** represents secure PC#1 and **GREEN** represents non-secure PC#2.

4.1.1 STEP #1: INSTALL SOFTMCCS ON BOTH PCS

Description:

Download and install softMCCS (http://www.entechtaiwan.com/lib/softmccs.shtm).

Notes: In a real world scenario, the malicious code (virus) is hidden.

Notes: The softMCCS application is a dedicated diagnostics and compliance application implemented on top of EnTech's next-generation, licensable softOSD communications library. Other MCCS capable applications are available on the internet. softMCCS was chosen due to its friendliness, which made it easy to demonstrate MCCS communication actions.
In a real-world scenario, an attacker would write malicious code to execute such commands.

4.1.2 STEP #2: LAUNCH THE SOFTMCCS ON PC#1

Description:

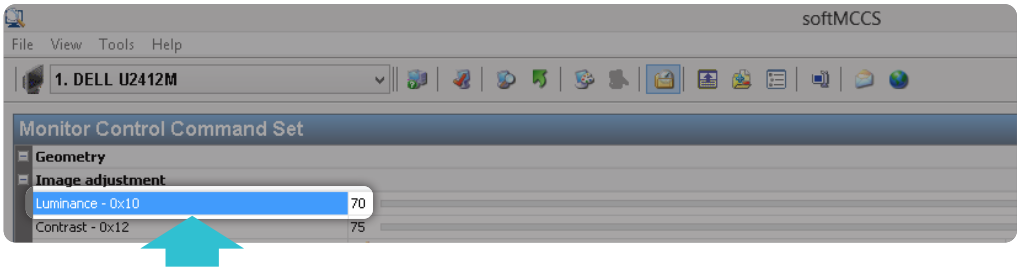
Click the softMCCS program icon to start the application.



4.1.3 STEP #3: READ THE CURRENT LUMINANCE SETTINGS

Description:

Open the **Image adjustment** section to view the luminance settings.



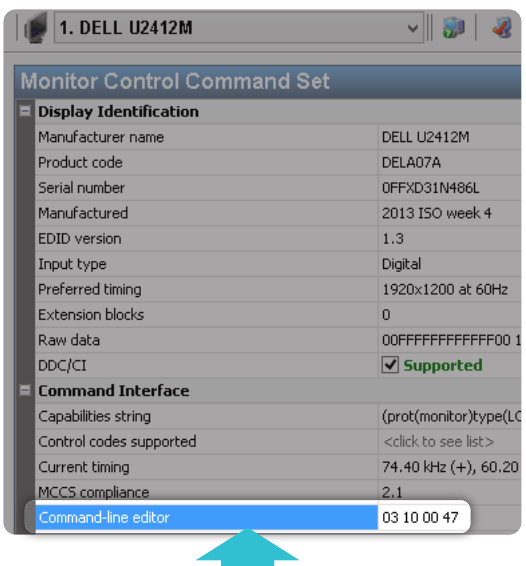
Luminance value = 70

4.1.4 STEP #4: CHANGE THE LUMINANCE VALUE TO 71%

Command:
Type: 03 10 00 47

Description:

Scroll down to the **Command-line-editor** and type **03 10 00 47.**
Click **ENTER** to apply the settings and change the monitor's luminance value to 71%.



Type: 03 10 00 47
Click ENTER to apply settings

4.1.5 STEP #5: RESCAN THE HARDWARE TO REFRESH THE READ OF THE CURRENT LUMINANCE SETTINGS

Description:

Click the **Rescan the Hardware** icon or press **Ctrl + F5** icon to refresh the displayed information.

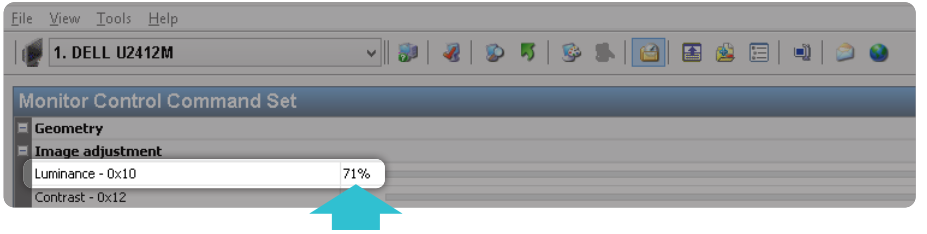


Rescan the Hardware (Ctrl + F5)

4.1.6 STEP #6: READ THE NEW LUMINANCE SETTINGS

Description:

Open the **Image adjustment** section to view the new luminance settings.



New Luminance value = 71

4.1.7 TASK ACCOMPLISHED

The new luminance value shows that the data was successfully transferred.

4.2 WORKFLOW EXAMPLE 2: MANIPULATE VALID CONTROLS TO SAVE NON-VALID VALUES

This procedure shows a step-by-step workflow to manipulate valid values with non-valid ones.

4.2.1 STEP #1: INSTALL SOFTMCCS ON BOTH PCS

Description:

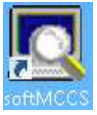
Download and install softMCCS (http://www.entechtaiwan.com/lib/softmccs.shtm).

Notes: In a real world scenario, the malicious code (virus) is hidden.

4.2.2 STEP #2: LAUNCH THE SOFTMCCS ON PC#1

Description:

Click the softMCCS program icon to start the application.



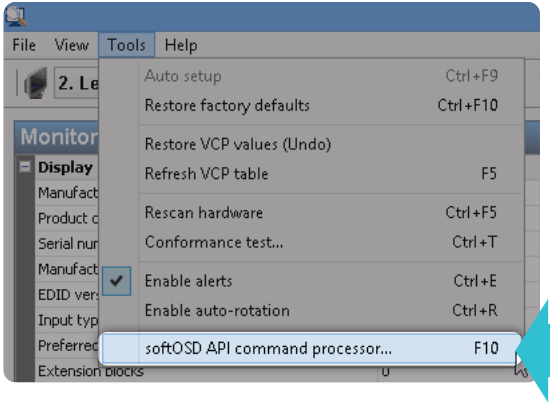
4.2.3 STEP #3: RUN A UTILITY TO QUERY THE MONITOR'S SETTINGS

Command:

Open the **softOSD API command processor** tool by selecting it from the Tools menu or pressing **F10**.

Description:

This tool reads the monitor control values.

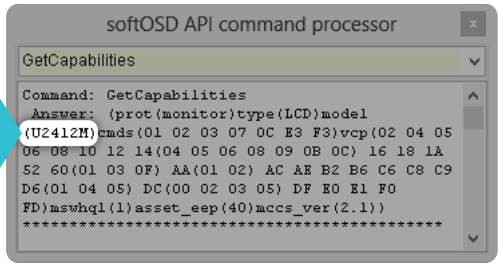


4.2.4 STEP #4: IDENTIFY MONITOR TYPE

Command:

Type: GetCapabilities

Monitor Type:
Dell U2412M



Description:

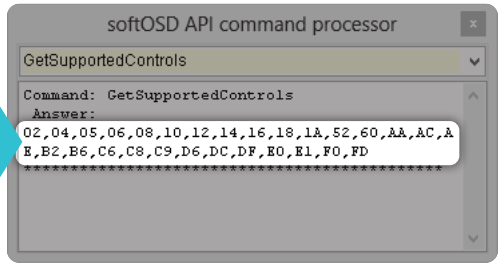
By identifying the monitor model, an intruder can discover the specific controls that can be attacked. Preparing an attack requires performing a preliminary study on various monitors.

4.2.5 STEP #5: QUERY SUPPORTED CONTROLS

Command:

Type: GetSupportedControls

List of
control
ID
numbers



Description:

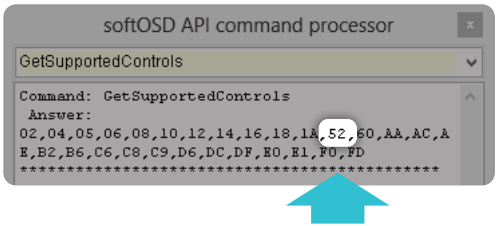
This command lists the controls that the monitor supports. Try replacing values for each control until you find one that accepts non-valid values.

Use this command as part of preliminary research before striking the target computers, OR use it during the attack to find available controls to store data.

4.2.6 STEP #6: SELECT A TARGET CONTROL TO EXPLOIT

Description:

Select control number 0x52



Last active control – 0x52

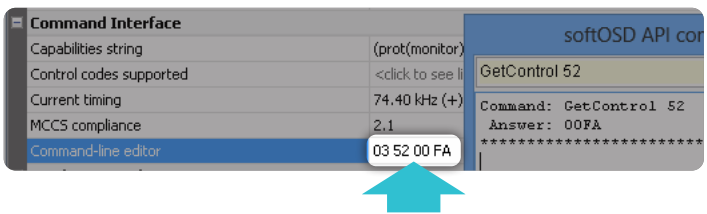
Notes:

- Select a target control to store data. Control 0x52 'Last active control' is chosen.
- Control 0x52 records the ID of the last control changed in the system. For example, after changing luminance settings, which are saved in control number 0x10, the output of control 0x52 is '0x10', as this is the ID of the last control that was changed.
- Control 0x52 is used for internal action validation and history. For this purpose, it is both readable and rewritable, and can store data in various formats (digits and letters). Changes to this control are unnoticeable to the user who is working with the monitor.

4.2.7 STEP #7: WRITE FALSE NON-VALID DATA

Command:

Type: 03 52 00 FA



Change the value of control 0x52
to FA where “FA” is false data

Description:

Change the value of control 0x52 to FA, where “FA” is false data.

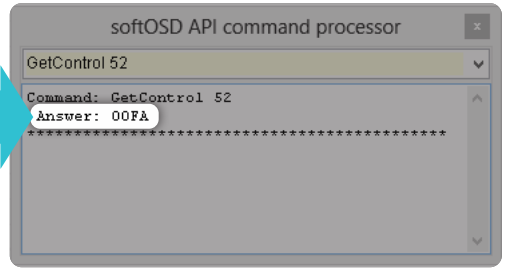
- Command-line editor is a tool in the softMCCS application, which sends MCCS commands to the monitor. In this document, the Command-line editor writes non-valid values into control 0x52. It shows how simple it is for an intruder to write an application that communicates with the monitor through MCCS commands over the DVI/HDMI cable.
- ‘Seek and Try’ to find non-valid control values. Enter the command: ‘GetSupportedControls’ to find all the available controls, described in step 5. Try writing non-valid values, such as F1, F2, F3…F9 and FA, FB, FC…FE, to each control. Once successful, the monitor is breached and can store data.

4.2.8 STEP #8: READ THE FALSE VALUE

Command:

Type: GetControl 52

New value data = FA



Notes: Read the current value of control 0x52.

4.2.9 TASK ACCOMPLISHED

The new value shows that the data was successfully transferred.

5. HOW HSL'S SECURE KVMS SECURE THE DIGITAL VIDEO COMMUNICATION

5.1 OVERVIEW

A secure KVM must fully isolate all coupled hosts from each other AND from the peripherals that are being shared through it, preventing host-to-monitor and host-to-host direct access. The KVM isolates the shared peripheral, such as a monitor, which can't be used as a broker for shared information.

HSL's secure KVMs achieve the highest level of security by enforcing all of the following strict mechanisms:

- EDID Emulation
- No shared circuitry
- Block MCCS
- Individual and isolated EDID layout for each PC

5.1.1 EDID EMULATION

To avoid direct attacks, HSL's secure KVMs isolate the monitor from its coupled PCs by emulating the EDID. During the boot process, the KVM scans the monitor's EDID content for authenticity and writes it into an isolated EDID EEPROM chip for each port. Each chip, which is isolated from the other EDID chips, emulates the EDID content without affecting the monitor's memory.

Each PC accesses a dedicated read-only EDID emulation. This enables its operating system and applications to operate smoothly, with no direct or shared access to the monitor, and thus does not compromise the monitor's security.

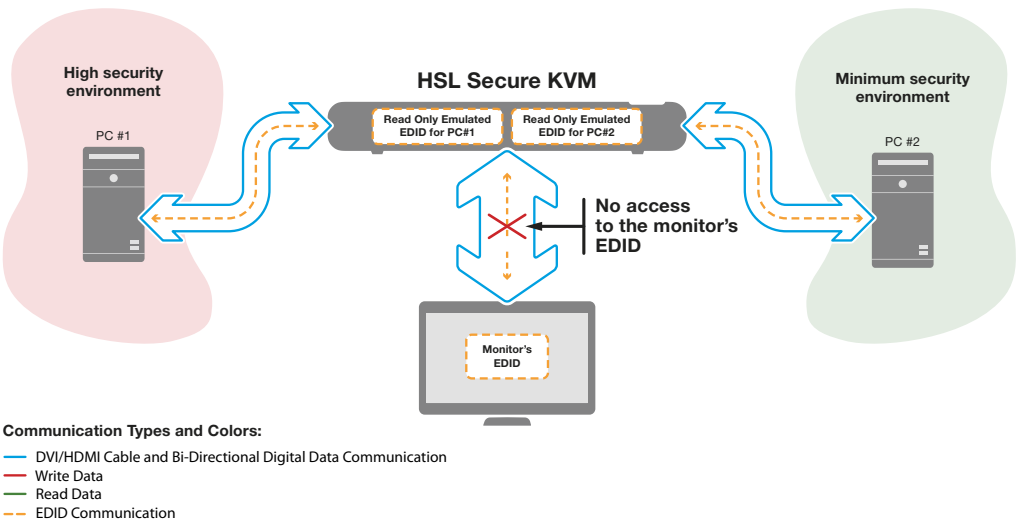


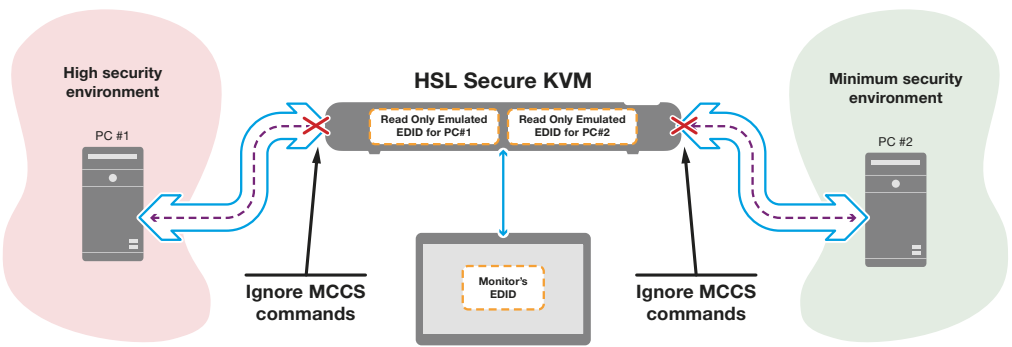
***Figure 7**. EDID Emulation*

5.1.2 NO SHARED CIRCUITRY

All EDID host computer channels are fully isolated from each other. The video switching module is designed with no shared circuitry or components across EDID interfaces. The basic module design enforces this isolation and protects the KVM from known or unknown threats that rely on using shared circuitry.

5.1.3 BLOCK MCCS

The KVM ignores MCCS commands, so they cannot pass through the KVM and exploit the monitor's internal memory. The EDID emulation EEPROM devices also have write protect switches, which isolate each individual PC.



Communication Types and Colors:

- DVI/HDMI Cable and Bi-Directional Digital Data Communication
- Write Data
- Read Data
- EDID Communication
- MCCS Commands
- Ignore MCCS commands

5.1.4 INDIVIDUAL ISOLATED EDID LAYOUT FOR EACH PC

The secure KVM has an individual isolated EDID emulation EEPROM for each channel. Each coupled host computer can only read its respective EDID emulation EEPROM device. Any attempt by an infected host computer to affect the memory on the EDID emulation EEPROM device is limited to that computer, and cannot propagate to any another EDID channel.

WWW.HIGHSECLABS.COM

High Sec Labs (HSL) develops high-quality cyber-defense solutions in the field of network and peripheral isolation for protecting national assets and infrastructure.

The company, headquartered in Caesarea, Israel, was founded in 2008 and has a second manufacturing site in the United States.

High Sec Lab's roots are deeply embedded in the defense and commercial industries developing cyber protection solutions. Among its customers are some of the world's leading governments and defense organizations as well as commercial companies such as banks, healthcare providers, and national infrastructure companies.

HighSecLabs
Highest Security Solutions

© 2025 HighSecLabs Ltd. All rights reserved. All trade names are trademarks or registered trademarks of respective manufacturers listed. Specifications are subjected to change without prior notice.
Patents: www.highseclabs.com/patents/. HLT33874 Rev 1.0