CHERI DE4 Getting Started Guide

Version 1.2

This interim document is not released for public consumption

Brooks Davis
SRI International and the University of Cambridge*

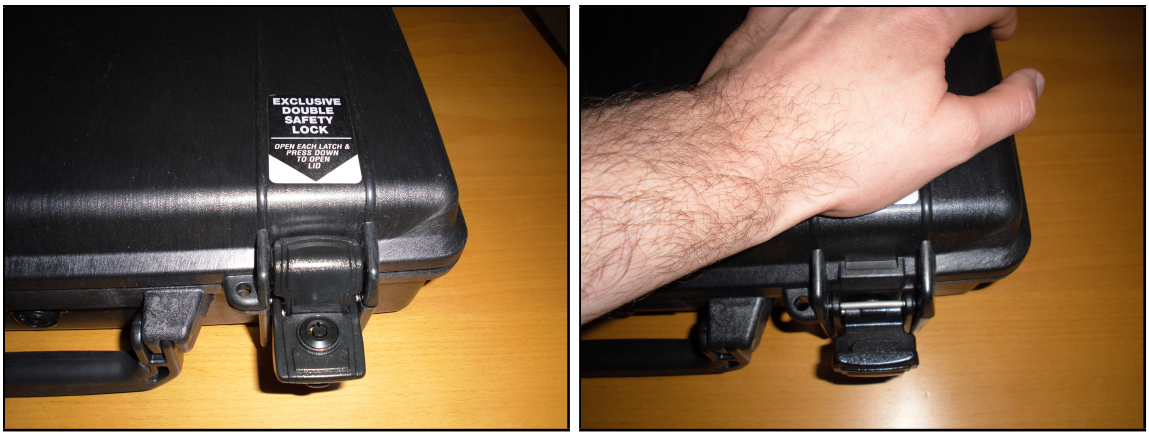
March 16, 2013

Abstract

This document is a getting started guide for SRI International and the University of Cambridge's demo platform for the research prototype implementation of the Capability Hardware Enhanced RISC Instructions (CHERI) instruction set architecture (ISA). The demo platform uses Terasic's DE4 FPGA board and their MTL touchscreen to implement a (rather bulky) tablet computer. The document is intended to document the state of demonstration software and provide enough information for a user to operate and and upgrade an assembled and pre-configured board.

The Getting Started Guide is targeted at demonstrators who will be displaying the current state of our demonstration hardware to interested parties. Instructions related to advanced configuration and/or development can be found in the *CHERI User's Guide*. This guide includes an overview of the relevant portions of the physical hardware as well as a review of the features of demo software.

*Sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-10-C-0237. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.



(a) Latch half open (b) Latch open

Figure 1: Opening the Pelican case

1 Unpacking the DE4

DE4 demo hardware is shipped in a Pelican 1470 hard side brief case. Unlatching the case is a two step process. First, the latches must be lifted as shown in Figure 1a. Then you must press firmly on the top case to allow them to be lowered, releasing the secondary lock as shown in Figure 1b.

Inside the case you will find a DE4 board with MTL LCD touchscreen in a plexiglas case, a power brick, a power cord, and an HDMI-to-VGA adapter. You may also have been shipped a USB cable and SD Card(s). See Figure 2 for a typical layout. After removing the DE4 from the case, check any easily accessible screws and bolt for tightness as they have a tendency to loosen in transit.

2 DE4 Demo Hardware

The DE4 FPGA board is covered in an intimidating array of connectors, buttons, and switches. Fortunately, only a few of these are required for demo operation. Figure 3 shows the back of the board as packaged. The battery input connector, power switches, and CPU reset button are the only things you will need to interact with.

To set up the board for the demo verify that the SD Card slot (shown in Figure 4) is empty[1] and that the various switches and DIP switches in Figure 6 are in the factory default down position (towards the PCIe edge connector on the other side of the board.) Then connect the power cord to the power brick and plug the power brick into the battery input connector which is visible in Figure 5. For general use, the system should be fully charged before any demonstration will all four charge indicator LEDs glowing solid blue.

With all the connections made, the power switches can be turned on and the system will boot. Booting takes approximately 90 seconds in the current configuration. During the boot process the touchscreen goes through a number of transitions. It starts green when the FPGA image is initialized. Once the kernel beings to probe devices it turns black. After startup scripts begin to run a screen showing “Booting CheriBSD Please Wait” is displayed. At the end of the startup process the screen again goes black and then a busy indicator is displayed while the demo application is loaded. Once the CTSRD logo is displayed the demo is

[1] SD Cards are only used for upgrades

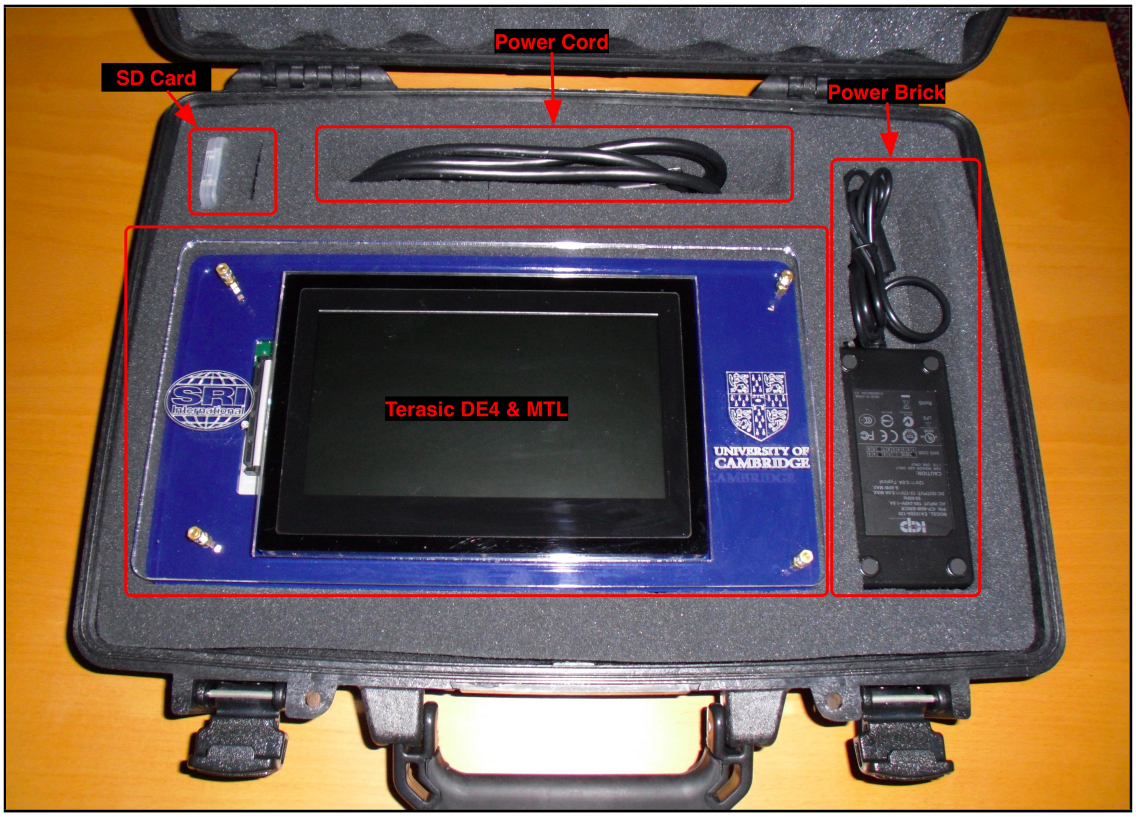


Figure 2: Inside the Pelican case

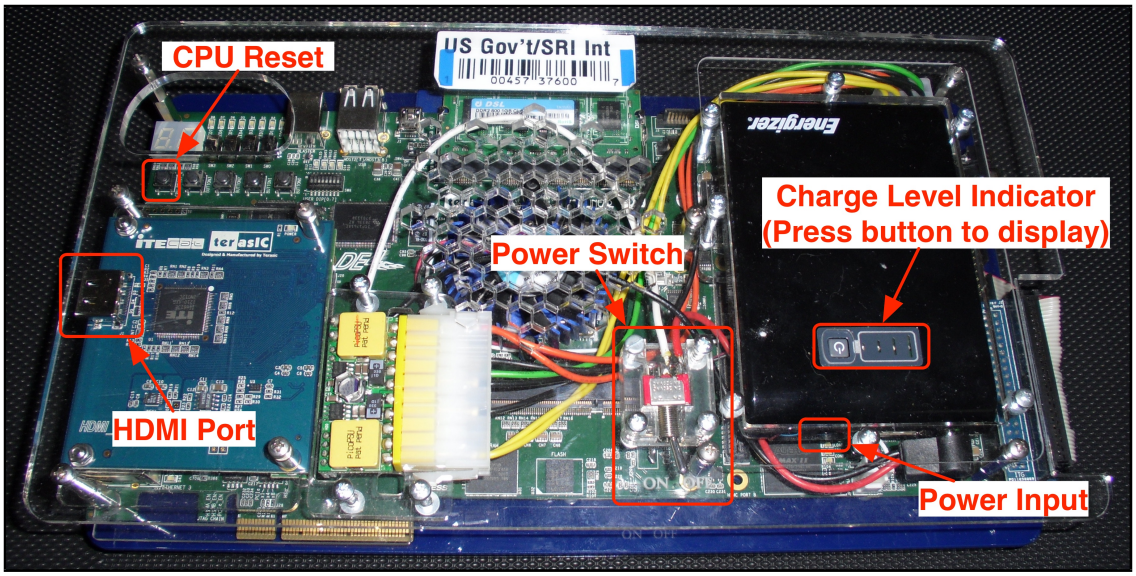


Figure 3: Back of the demo DE4 package

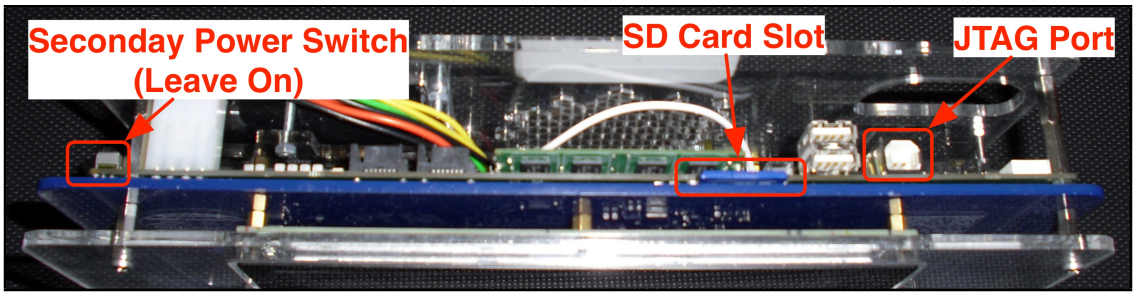


Figure 4: Top of the demo DE4 package

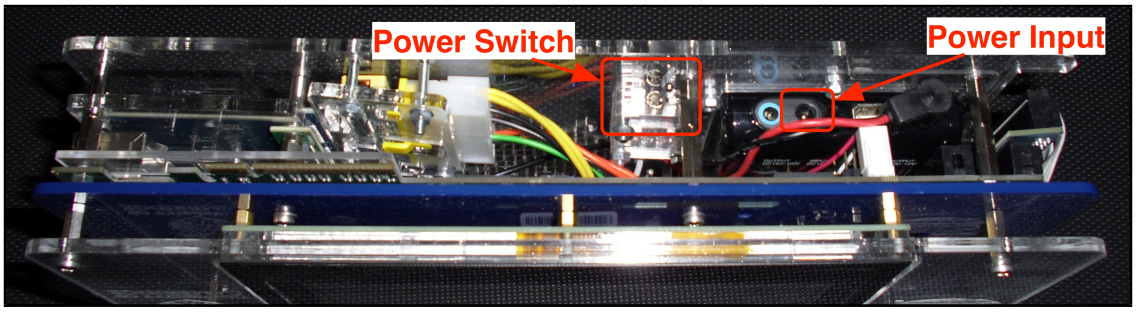


Figure 5: Bottom of the demo DE4 package

ready to be used.

When you are done with a demonstration, switch the rear toggle switch to the off position to prevent further drain on the battery. Even with the rest of the system off, the power supply will drain the battery in a few hours unless this is done. The secondary power switch (slide switch on the DE4 board) may be safely left in the on position.

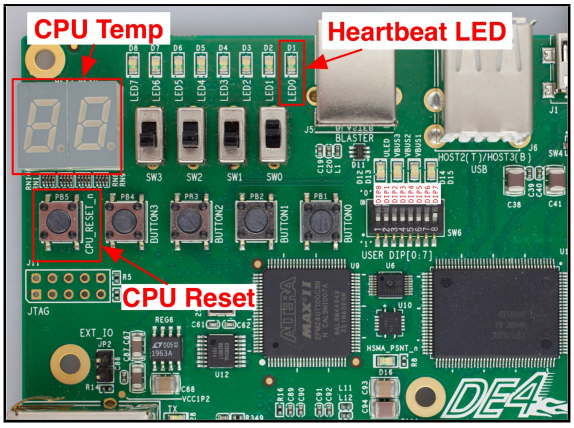
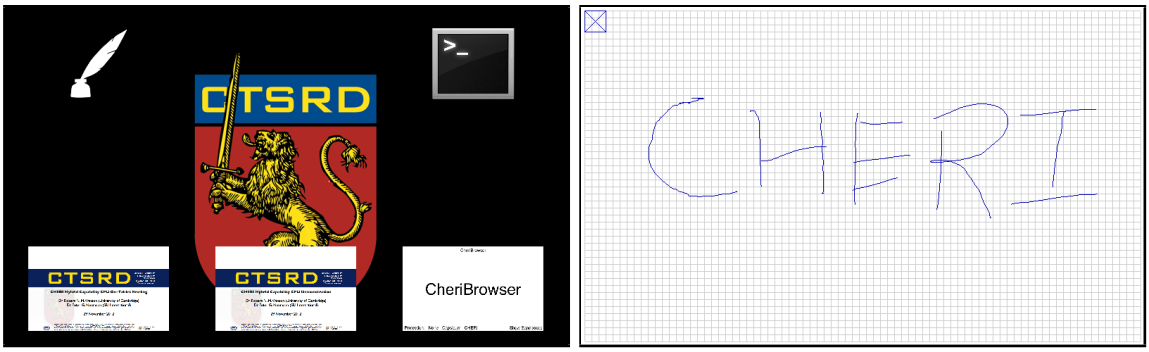


Figure 6: DE4 Buttons and Switches

2.1 HDMI Output

The HDMI port on the DE4 daughter card outputs a 720x480 image of the left most 720 columns of the 800 column touchscreen display. When connected to a display using an HDMI or HDMI-to-DVI cable the full area should be displayed and the display will be sharp. When using an HDMI-to-VGA adapter we find that either the left or center 640 columns of this region are displayed at 640x480. In some cases the pixels are mis-sampled resulting in a somewhat blury picture. Which case seems to depend on the display and in some cases varies between plug events. Currently, only the CheriPoint presentation tools supports multiple resolutions.



(a) Main Screen (b) Drawing Program

Figure 7: Pictview Screenshots

2.2 SD Card Input

The demo software image has been updated to mount FAT formatted SD Cards automatically at `/sdcard`. Content on the SD Card may be accessed using CheriBrowser. While we have conducted limited testing of removal of the SD Card, it would be well advised to close any programs using data on the card and navigate away from the `/sdcard` mount point before removing the card. More information on CheriBrowser may be found in Section 3.4. Information on constructing CheriPoint slide decks may be found in Section 3.3.

2.3 DE4 Buttons and LEDs

One additional note about the hardware. After the kernel has probed devices `LED0` will blink approximately every 900ms as a system heartbeat. If the system crashes the LED will stop blinking. Note that the light may stop blinking for a significant period of time during OS upgrades. In that case the system can be rebooted by power cycling or by pressing the `CPU_RESET_n` button. Both `LED0` and the `CPU_RESET_n` button are marked in Figure 6.

3 The Demo Software

The main demo screen (see Figure 7a) has four options.

3.1 Drawing Demo

A quill icon in the upper left corner leads to a simple drawing program as shown in Figure 7b. Pictures can be drawn with your finger and erased by touching the X in the upper left hand corner. The drawing function may be exited with a pinching gesture.[2]

3.2 Terminal

A terminal icon on the upper right corner starts a shell than can be accessed via a virtual keyboard. The demo platform runs a very stripped down set of commands, but simple things like `ls` and `top` work as does `vi`. In addition to the output of commands run via the shell, the output of other subprocesses such as

[2]Gesture detection is not perfect so you may need to pinch multiple times to exit a function or program.

the CheriBrowser and its children may be displayed on this terminal. The terminal may be closed with a pinching gesture.

3.3 CheriPoint

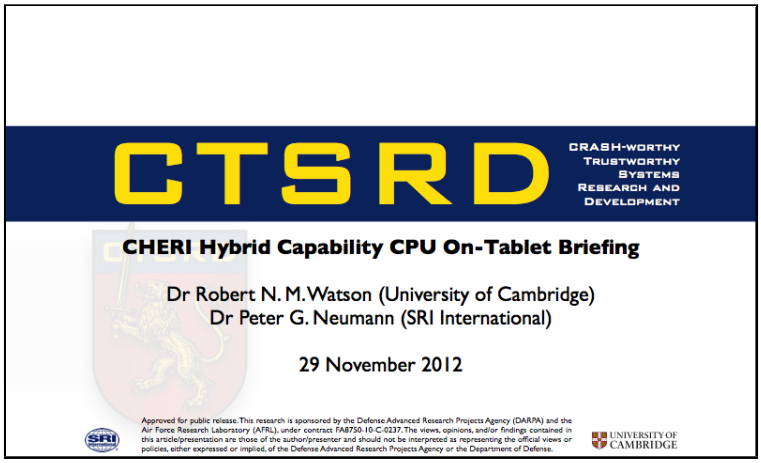


Figure 8: Default view

A presentation title slides in the lower left corner and lower center lead to a slide viewer. On the demo image, the lower left slide is a short briefing about the CTSRD project and the lower center one is a demo briefing of Capsicum and CHERI protections.

The presentation can be advanced by swiping from right to left. Previous slides can be viewed by swiping left to right. The image viewer may be closed with a pinching gesture.

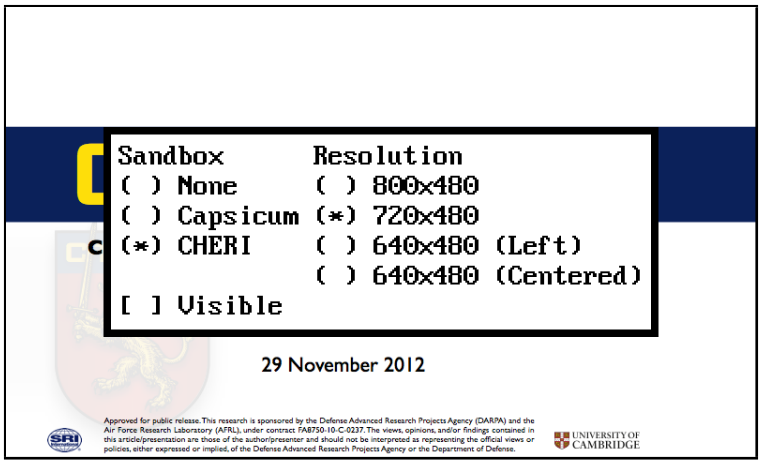


Figure 9: Default view

A configuration dialog as shown in Figure 9 may be accessed with an upward swipe. It can select which sandboxing model is used, toggle visibility of sandbox boundaries, and adjust the screen to accommodate different output resolutions. Note that both the radio button or check box elements and the text describing

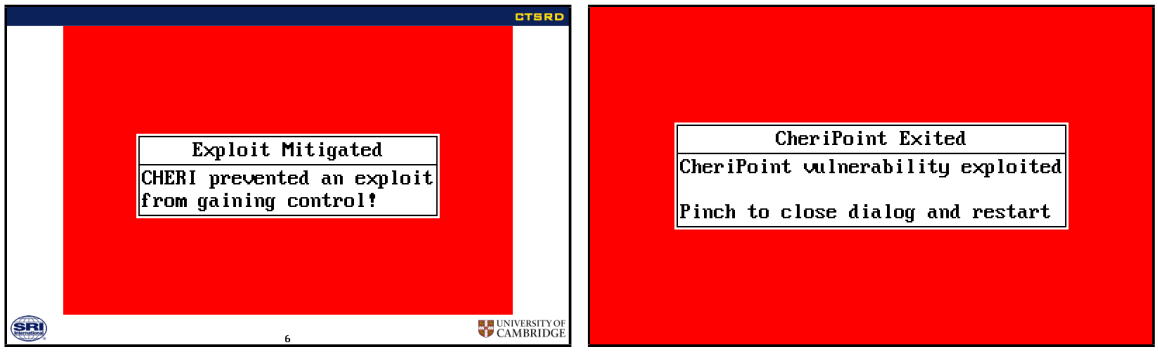
them may be selected and the text may be easier to select. The configuration dialog may be dismissed with a pinch or downward swipe gesture.

The briefing slides are generally self explanatory and can be presented with no configuration other than adjusting the resolution for the projector.

Figure 10: Default view

3.3.1 Demo: Trojan horse in `libpng`

The demo requires more explanation. Slide 2 provides a bit of information on the demo platform. Slide 3 sets the stage with a discussion of the process of assembling talks on large projects where multiple people submit slides or images. Those people many not be entirely trustworthy and with CheriPoint you don't have to trust them. Slide 4 discusses the architecture that makes this possible along with the nature of the `libpng` trojan horse. Slide 5 is a typical contributed slide.



(a) With CHERI sandbox (b) With no sandbox

Figure 11: Viewing the trojan triggering slide

Slide 6 contains the actual trojan trigger. When the trojan is triggered the behavior will depend on the current Sandbox mode. If the sandbox is in CHERI or Capsicum mode then `libpng` will successfully fill the image buffer with red and fail to run the exploit code. CheriPoint will detect that the exploit was attempted an report it as shown in Figure 11a. The report dialog can be closed by pinching. Alternatively, the next or

previous slide may be accessed with an appropriate horizontal swipe. In the case of no sandbox, the trojan will trigger resulting in an external program running and washing the screen red and killing the process which causes the monitoring process to display a dialog as shown in Figure 11b. When the dialog is closed, CheriPoint is restarted on the last slide that was successfully rendered.

Slide 7 shows the architecture of the sandboxes. The presenter may wish to use the configuration dialog to toggle the sandbox indicators on as shown in Figure 10. With repeated toggling of the sandbox type it is possible to see that CHERI is faster than Capsicum due to the decrease in length of the pauses between the rendering of each element.[3]

The trojan works as follows. PNG files are a series of named and checksummed data segments called chunks. We have added support for a new chunk type to `libpng` named `exEc`[4]. When an `exEc` chunk is found, it is parsed for an embedded command and set of arguments which is then executed. An unpublished program called `execpng` can be used to insert a such a chunk into any PNG file.

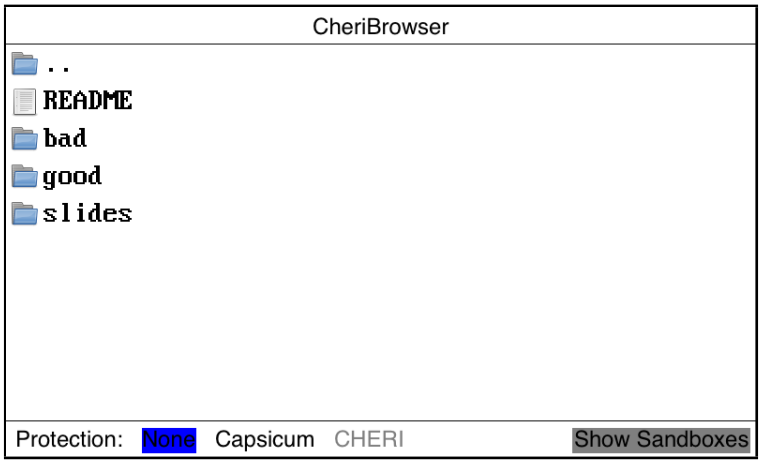


Figure 12: Default view

3.3.2 CheriPoint slide decks

With the introduction of SD Card support and support for opening presentations in CheriBrowser the demo platform now supports user supplied presentations. A presentation is a directory with the extension `*.cpt` that contains a series of `*.png` files.

Files that match the pattern `*-cover-*.png` are considered covers and one will displayed as the first slide of the presentation as the only image on the screen, centered relative to the configured display area. If the image is narrower than the screen the left and right most pixels of each line are extended to the respective edges of the screen to provide a more finished look. If multiple covers are present then CheriPoint will attempt to find one that is formatted for the width of the screen. For example, if the screen is configured for 640x480 display, the code will first look for a cover matching `*-cover-640.png` before falling back to the one that is lexically first. In most cases it is simplest to provide only a 640x480 cover image.

All other images are displayed in lexical order. If the image is the full height of the screen (480 pixels), then it will be displayed similarly to the cover slide. Otherwise it will be composited with a CTSRD header,

[3] We intend to provide a quantitative report of the sandbox overhead in a future version of CheriPoint once we add the required CPU features.

[4] The capitalization is an artifact of the PNG format

SRI International and Cambridge University logos in the lower left and right corners, and a page number in the bottom center. The slide body will be rendered centered relative to the display area starting immediately below the header. The largest usable size for such images is 640x410. Be advised that a small portion of the lower left hand corner will be obscured by the SRI logo.

3.4 CheriBrowser

In the lower right hand corner is a thumbnail of the CheriBrowser which is our security demo. CheriBrowser is a simple file tree browser than displays icons and file names for each file system object as showing in Figure 12. For regular files, the icon displayed is determined by using `libmagic` to examine the contents of the files in order to determine their mime-type. Using CheriBrowser users can navigate the file system at will and view files that `libmagic` identifies as plain text (`text/plain`) or PNG images (`image/png`) as well as CheriPoint presentations. Text file viewing is somewhat limited, but some vertical scrolling is implemented and it is sufficient to display the `README` file than discusses our demo hierarchy. Both text and PNG image views may be closed with pinching gestures as can the full application. CheriBrowser displays a CheriPoint icon for any directory ending in `.cpt` and invoked CheriPoint when the directory is selected.

The use of `libmagic` to determine file types is a common technique in file browsers and even security applications like virus scanners, but is inherently risky as it requires handling untrusted data, sometimes in complex ways. This has resulted in vulnerabilities in `libmagic` in the past.[5] On the demo platform, we have modified libmagic to introduce a backdoor than allows arbitrary code execution.

CheriBrowser is started in the `/demo` directory. This directory contains a `README` file explaining the demo and three subdirectories: `bad`, `good`, and `slides`. The former two subdirectories contain the same set of png files, but the `bad` directory contains an additional file that triggers the backdoor in `libmagic`. The file runs code than executes another program that washes the framebuffer with red. The `slides` directory contains the demo slides accessible from the main screen as well as a `credits.cpt` that contains more information about the CTSRD project and a number of photos of the team and the demo hardware.

(a) Without Sandboxing (b) With Capsicum

Figure 13: Viewing the `bad` directory

By default the exploit above is successful both washing the screen with red and taking over execution of CheriBrowser. Because we have run CheriBrowser in a wrapper we can detect the exit of the exploit code and display a dialog box indicating the exploit was successful as shown in Figure 13a. A more subtle exploit could take complete control of the application.

[5] `http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1536`

The interesting part of the demo is that we can enable Capsicum and (in the future) CHERI protections when analyzing files. This is done by touching the protection names at the bottom of the screen. In a small directory such as the `good` directory we can easily see the performance impact of using a process sandbox for each file by toggling back and forth between None and Capsicum modes. The value of Capsicum protection can be seen if we navigate the the `bad` directory. There we see a toxic icon for `badfile` if Capsicum protection is enabled as seen in Figure 13b and an application crash if not.

When in Capsicum or CHERI protection modes you can view the areas of the screen whose contents are derived form sandboxed data by tapping the "Show Sandbox" text in the bottom bar. This causes the icons that are selected based on data generated in the sandbox[6] to be outlined in yellow as showing in Figure 14.

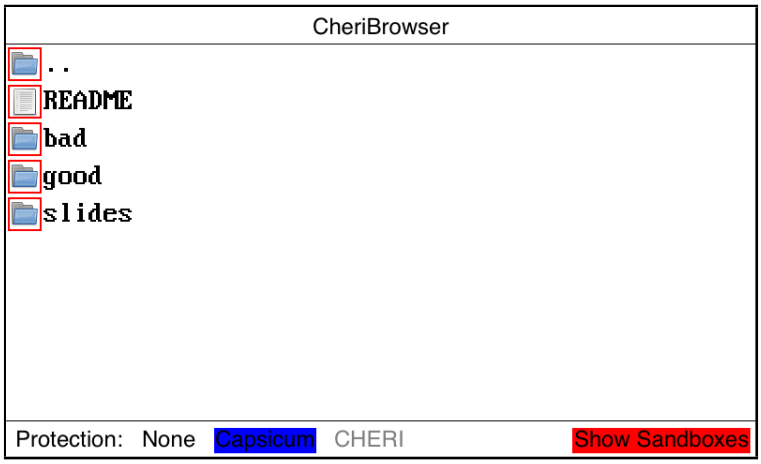


Figure 14: Sandboxes shown

Future editions of the demo will enable CHERI capability support that will allow us to provide equal or better protections to the Capsicum case without the expense of creating a process per file.

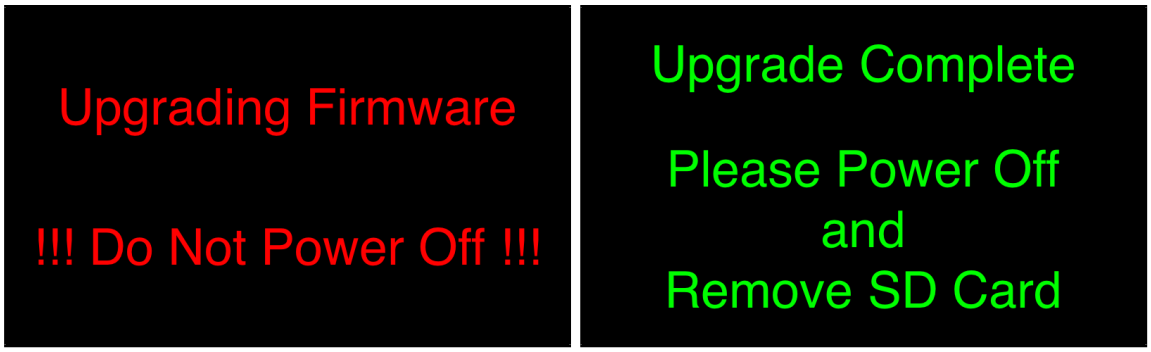
3.5 About Box

One final feature accessible from the main launcher screen is an about box showing the version of the CPU bitfile and the FreeBSD kernel. It can be accessed with a two finger swipe from left to right. It can be closed with a pinch or a swipe from right to left.

4 Upgrading the DE4

Upgrading the DE4 is a simple process. You just obtain an upgrade image on an SD Card or write one to an SD Card, insert the card into the board, and reboot. The system will boot and immediately proceed with upgrading. The touchscreen will display the image in Figure 15a while the upgrade is in progress and then will display the image in Figure 15b when the upgrade is complete. This upgrade process takes approximately 20 minutes.

[6]Technically speaking, only regular files or symbolic links pointing to regular files are actually processed in a sandbox. Because it is safe to do so and it is easier to contain the sandbox if it does not have filesystem access, we identify other types of filesystem objects such as directories and device nodes in the main application code.



(a) In Progress (b) Complete

Figure 15: Upgrade Status Screens

Once the upgrade is complete, the SD Card should be removed and the system should be rebooted to verify functionality. Current versions of the upgrade process are not smart enough to avoid erasing and reinstalling the firmware so leaving the SD Card in will result in the process repeating.

Take care that power is not interrupted during the upgrade process. It is strongly advised to ensure that the unit is fully charged before proceeding with an upgrade. Loss of power during the erase and write phases of the upgrade will require a full development environment for recovery and the process is modestly complex. If recovery is required, the process documented in the *CHERI DE4 Factory Install Guide* may be used to restore the unit to working order.