Embedded Capture-the-Flag (eCTF) – Q&A

1 Are hardware modifications allowed?

That depends… as part of your secure design, the answer is “no”. We cannot allow hardware modifications to be part of your design because it would become too difficult to share those modifications with the teams that will be attacking your design. The system handoff is an exchange of software images only. You can assume that your image will be installed on a widget that has the keypad installed and the program jumpers. You may add instructions to your installation process to remove the program jumpers, but keep in mind that an attacker can easily add those jumpers back.

During the attack phase, hardware modifications are allowed and encouraged.

1.1 Isn’t that unfair for the defense?

Yes, but that’s life. When designing a real-world system, custom hardware modifications are often expensive and/or don’t scale well. On the other hand, the attacker only has to make their modification once.

2 Do we need to mail or submit the physical device?

No, there is no need to mail or submit the physical device – students are expected to submit only the system image for the device. The system image should allow other teams to load that image onto their own devices which can be registered with the server. See section 3.4 of the challenge writeup.

However, during the on-campus visit (see section 4.3) there will be a physical exchange of devices.

3 Can we assume that any re-registration of the Widget will be performed in a secure environment? Rule #3 says that *initial* registration is guaranteed to be performed in a secure environment, but is re-registration afforded the same grace?

Yes, you can assume that your re-registration process will happen in a secure environment. In fact, we don’t intend to use re-registration unless we need to during the initial setup and for testing. The intent of the requirement is mainly to prevent teams from designing fragile systems that brick the hardware if something goes wrong on the first registration attempt. The re-registration feature also enables several testing scenarios that we imagined could be useful, which otherwise might not be possible without spending a lot of time re-imaging devices or buying a lot of extra hardware.

4 How closely are re-registrations validated by the server admin? Will the admin want to inspect the hardware and initiate a re-registration himself, or would he honor a re-registration if tenants did it themselves, remotely?

ALL registrations (initial and re-registrations) will always be closely validated by the server admin. In a real system, the server admin would NOT honor a registration if tenants initiated the process themselves. However, for this competition, we will allow the attacking teams to initiate registration themselves only because this greatly simplifies the logistics of getting each team a registered device. Rule #3 states that even though we are allowing teams to perform the registration, we do not want attacks performed prior to or during that time. We may ask team advisors to help with this process, but we hope we can trust in the honor-system here. Remember that each flag that your attackers might claim will need to be validated with a short description of their attack (Rule #4)… if they abused the registration process then we’ll invalidate the points.

That said, it is allowable for an attacker to initiate a re-registration in an attempt to compromise an OLD registration. There's a subtle difference here, which makes this attack allowable. In a real system, there's nothing to prevent the tenant/attacker from stumbling across the re-registration functionality. If they do that, the worst-case scenario should be that the landlord needs to ask for the Widget back so that the re-registration can be done in a secure environment – until that happens, the tenant might be locked out… their own fault for messing around… so that design is OK.

However, beware of this scenario: an attacker initiates a re-registration and the Widget or server assumes that it's in a secure environment and reveals secret information which can be used to compromise OLD registrations (or even future registrations). That would be a bad design, and an attack that exploits it will be allowed (Rule #6 applies here).

5 Do PINs change upon re-registration?

It's up to you as the designer. In our example code, we're doing the PIN checking on the server-side, so we have a default PIN that is assigned by the server for all new registration requests. It's fine to do it differently if you want. I think setting a default PIN keeps things simple, but in some cases (for example, if you do PIN checking on the Widget) it might be easier to leave the PIN unchanged. Either way is fine. The master PIN provides a way for the admin to always be able to get the PIN back to something they know.

6 What's the best way to create our system image of the BBB?

There are many ways to do it (google!), but here's how we typically do it:

Boot from an SD card that has been setup with one of the default images (http://beagleboard.org/latest-images) and then plug a USB thumb-drive into the USB port on the BBB and use 'dd' to copy the eMMC to a compressed image onto the thumb-drive:

```
dd if=/dev/mmcblk1 bs=16M | gzip -c > /media/1274-BA99/bbb.img.gz
```

- Note: the USB drive isn't always mounted to /media/1274-BA99. The output of 'dmesg' will help you determine the device for the USB drive, then 'df' or 'mount' will tell you where it's mounted. Usually when you plug in a USB thumbdrive to the BBB it will automatically mount it to someplace under /media, but this might not always be the case.
- Note-2: You should also verify that /dev/mmcblk1 is the correct device for the eMMC on your BBB after you boot from the SD card. It probably is, but it might also be the SD card and the eMMC could be at /dev/mmcblk0. You can use "fdisk -l /dev/mmcblk1" to list the partitions on that device and you may be able to use that to differentiate between the eMMC and the SD card that you're booted from.
- Note-3: You also don't really need a USB thumb drive to do this. The SD card that we provided is more than large enough to fit the image of the eMMC. However the partitions on the SD card (as you currently have them formatted) might not be large enough without resizing them. Resizing partitions can be a pain, but you can easily add an additional partition to use just for storing the image file of the eMMC. The USB thumbdrive might be quicker and easier, so we recommend starting with that.

7 The “No Lockout” requirement says that we must allow a rate of at least 60 incorrect PIN attempts per hour. Are there any additional restrictions on how we implement the rate throttling (e.g. exponential backoff, etc.)?

No. There’s no need to make this complicated. As long as your design doesn’t have permanent lockouts and allows 60 incorrect PIN attempts per hour, it passes the requirement.

8 Is social engineering in-scope for this competition? Can we send phishing communications to other teams to trick them into revealing their secrets?

No, please don’t do this. Keep your attacks technical. ☺ We love creative ideas, but this one can easily violate university, state, and federal regulations.