Embedded Capture-the-Flag (eCTF)

1 Challenge Description

You're a landlord and you're tired of changing the locks on your rental property every time you get new tenants. The obvious solution (to any engineer) is to go digital and build an Internet-enabled door lock! How hard could it be? **Your challenge is to design and implement a system to unlock the front door that utilizes two-factor authentication**: that is, authentication based on (1) something you have, and (2) something you know.

- *Something you have:* The unlock device (also known as the "Widget"), is a physical device that acts as the user interface to unlock the front door. You will build this using the BeagleBoneBlack (BBB) + CryptoCape (CC).
- *Something you know:* This is a 6-digit personal identification number (PIN) that the tenant/user enters into the Widget.

The Widget will be given to your tenants to gain entry into the apartment. Given that your tenants are engineering students, they are likely to want to work out exactly how the units work. Some may even decide to cheat the system (e.g., emulate the Widget on their cell phone so they don't have to carry it, give a copy of the device to their friends or short-term rental customers, make their own device to get into the rental property after the lease is expired, lock out a roommate). **In other words – your tenants are possible attackers!**

Your system must meet a set of requirements (below) and should defend against as many attacks as you (and the other teams) can think of. You must design and implement both the Widget and the server to which the Widget authenticates. Once your system is completed, it will be subjected to attacks from the opposing teams, while you get a chance to attack the designs from the other teams. To set the ground rules, it is assumed that attackers have physical access for an extended period of time with the Widget, but the server is locked away in an area of the house that is inaccessible. Therefore, physical attacks on the Widget are fair game, but only remote attacks are permissible on the server. *The purpose of this scenario is to encourage a focus on security for the embedded system (the Widget) and to gain a practical understanding of ALL types of attacks.*

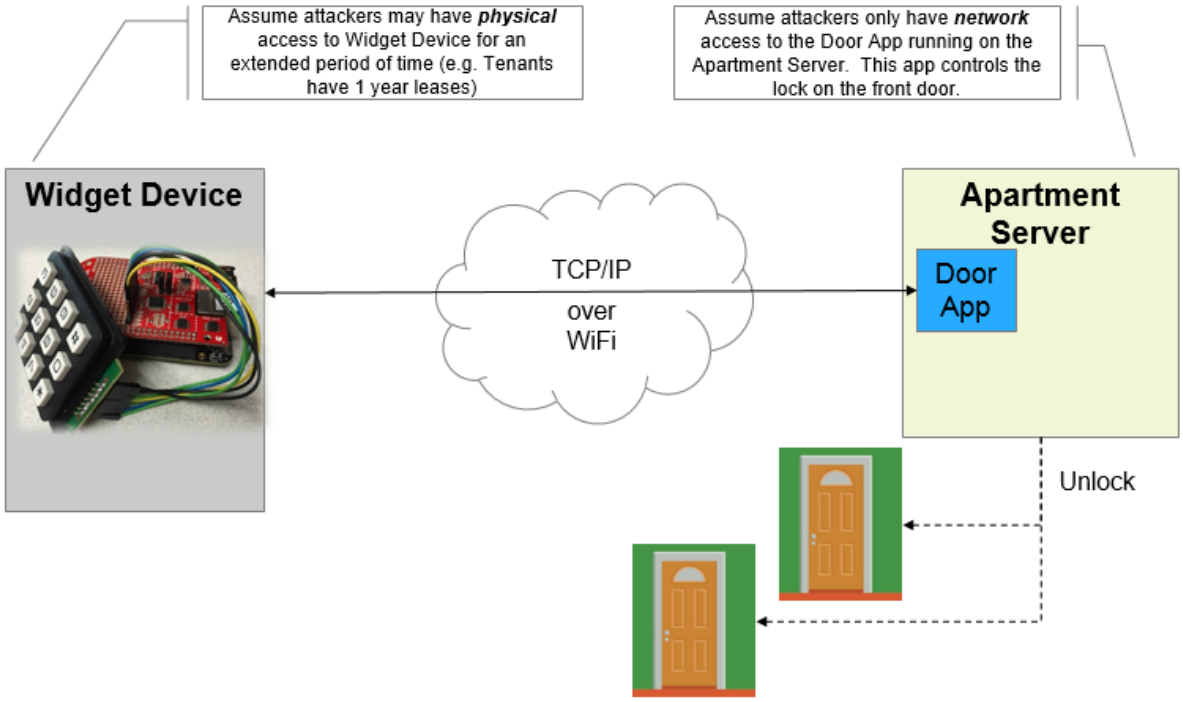

Figure 1. Challenge System Architecture

2 Phases

The eCTF is composed of the following phases:

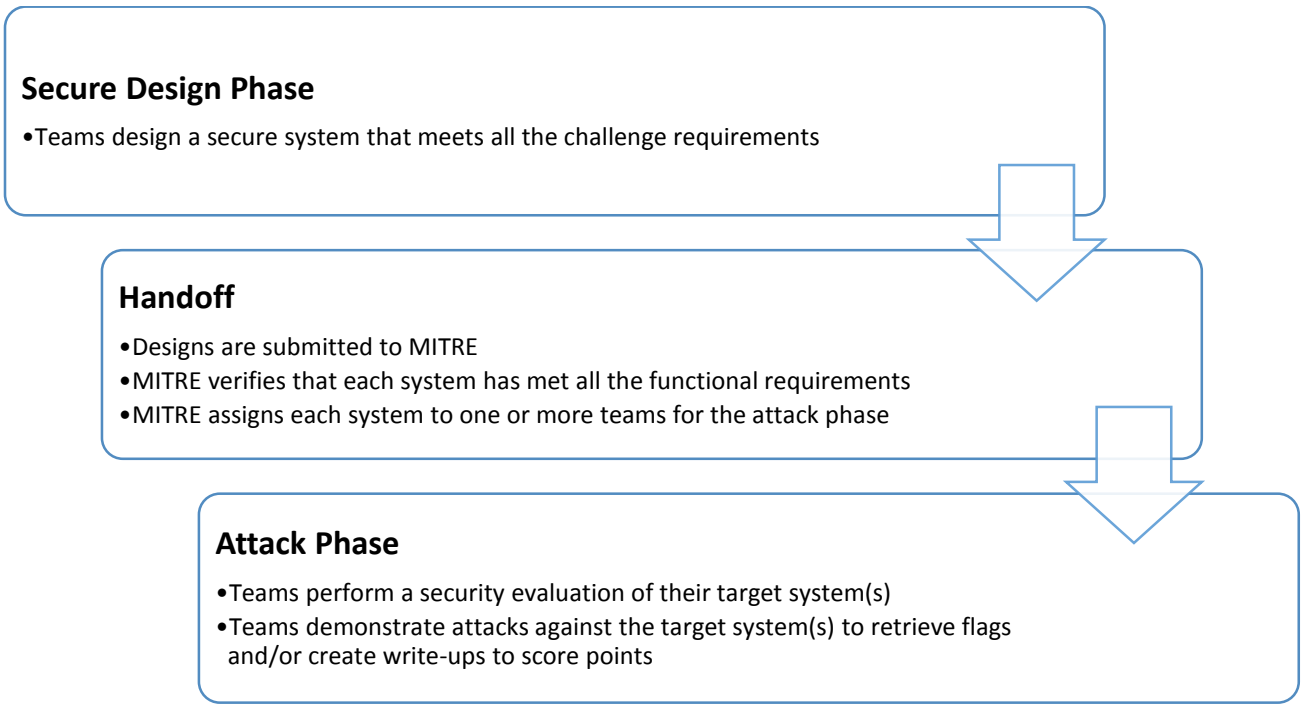

Figure 2. eCTF Phases

2.1 Schedule

***2016.01.13*: Begin Secure Design Phase**

Challenge details released.

***2016.02.24*: System Handoff**

Secure designs are due and Attack Phase begins.

2016.0x.xx: On-campus Visits

MITRE eCTF Admin visits each team.

***2016.04.13*: Competition concludes**

Write-ups are due and live-scoreboard stops accepting new flags.

***2016.04.15*: Award ceremony and debrief**

All teams come together to debrief and congratulate winners.

3 Secure Design Phase

During this phase, your team will design and implement a system that meets the functional requirements below and defends against attacks from the tenant or malicious outsiders (see Sections 3.3.3 and 6.1). Start by reading and understanding the functional requirements, common attacks, and flags, and then begin designing your system.

Keep in mind that the purpose of the Widget is to provide two-factor authentication. Therefore, the Widget device must be a factor in the authentication process. Knowledge of the passcode alone should not be enough to unlock the door – the tenant should need the passcode *and* the device. As the landlord, when you get the Widget device back after a lease expires, you want to be confident that the previous tenant can no longer get in.

3.1 Skeleton Example / Reference

MITRE will provide an example skeleton system that you can use as a template to begin building your system. This reference system is completely lacking of any security measures, but meets the functional requirements for the system. You may add your security features to this existing code, reuse components of the skeleton, or start from scratch.

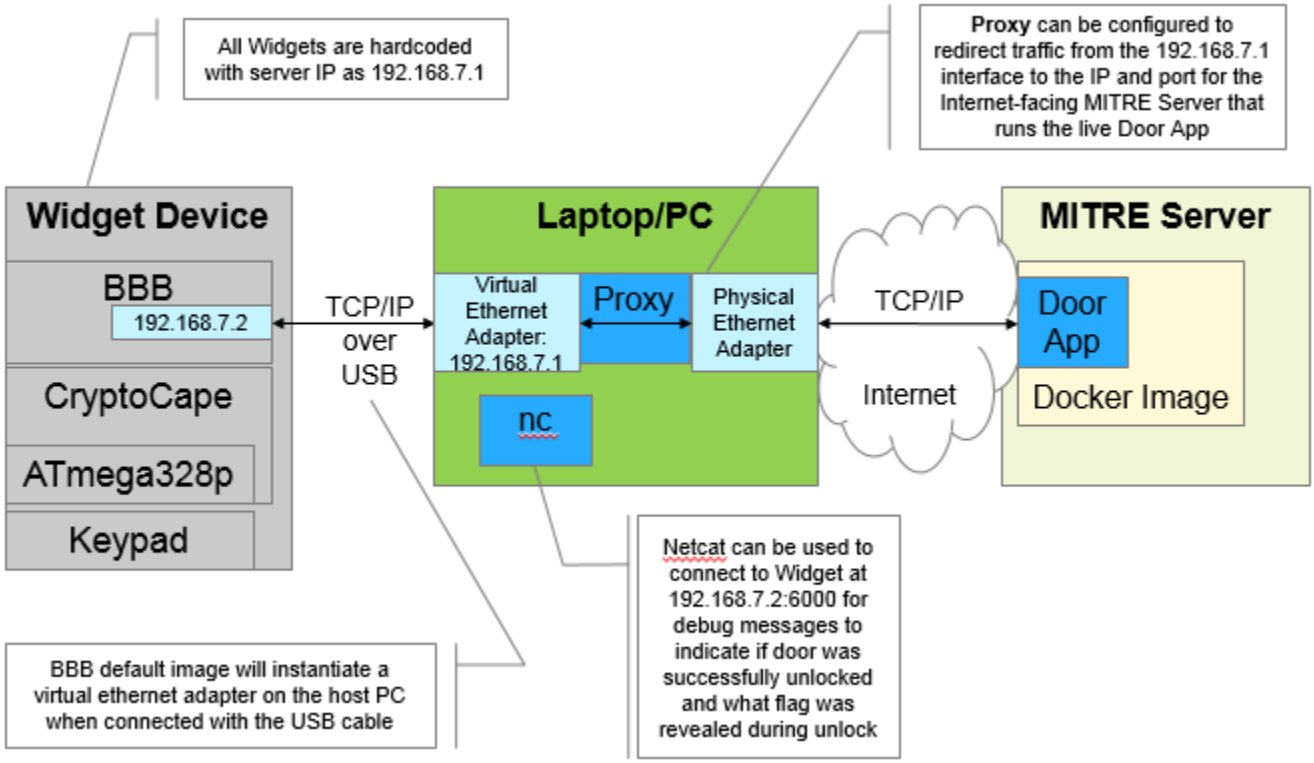

Figure 3. eCTF Setup

3.2 Functional Requirements

Name	Requirement
Widget	
Image and Setup	Each team must supply a system image for their BBB which upon **first** boot shall configure the BBB and CC if necessary (i.e., a user should not have to take any separate steps for initial configuration - everything should be automated). Note: Configuring the CC *may* be a one-way process (i.e., once it is configured for one implementation, it may not be possible to install a different team's implementation due to the way that the crypto modules on the CC may be used).
Fixed Destination	All server requests (unlock, register) are sent to the fixed IP address of 192.168.7.1, TCP port 5000. These requests will be forwarded to the actual server by the Proxy software.
Registration	Entering the special code: `*#*#*#*#` will initiate a registration request. Implementation of the registration request process and data formats are up to the implementer.
Re-registration	A Widget that is already registered can register again with a new server (or the same server) without needing to be re-imaged and without making any hardware mods to the CC. Note 1: This requirement is designed to prevent a lot of "one-use" CryptoCape hardware. Note 2: Depending on the system design, this may break the Widget's previous registration.
Unlock	Entering a 6-digit tenant PIN followed by `#` initiates a door unlock attempt, which is sent to the Door App server.
Visual Status	Device visually (i.e., with LEDs) indicates successful or unsuccessful door access and logs the result of the attempt (with flag string, if successful) to any connections on TCP port 6000.
Change PIN	Allow the tenant to change their PIN by entering: `<Current 6-digit tenant PIN>*<New 6-digit tenant PIN>#` Note: This could be Widget or server based – up to implementer.
Master PIN	Allow the landlord to change the tenant PIN by entering: `<8-digit master PIN>*<New 6-digit tenant PIN>#` Note: This could be Widget or server based – up to implementer.
Door App / Server	
Registration	**Widget-Registration-Data** resulting from registration requests will be appended to the **requested-widgets.txt** file. The **Widget-Registration-Data** can be any data structure implementers would like (crypto keys, id strings, PIN values, etc.). To accept a registration request, a server admin will copy the appropriate line of data from **requested-widgets.txt** to **registered-widgets.txt** and modify as needed (e.g., to add a corresponding "flag" if applicable, etc.) and then restart the server. The specific contents of the **Widget-Registration-Data** are up to the implementer, as long as necessary modifications and the addition of flags can be completed using a basic text editor.
Multiple Doors	The server must support multiple registered Widgets (i.e., the server should be able to uniquely identify and authenticate different physical Widgets). During application startup, all registered Widgets are read from a config file (the **registered-widgets.txt** file**).** Note: This allows easy configuration for other teams and the eCTF administrators.
Unlock Response	Server responds to all unlock attempts with success/failure indication and (if success) the "flag" from the **registered-widgets.txt** file. This file contains all **Widget-Registration-Data** entries and flag values (ASCII strings). This file is intended to be updated manually by the server admin. An example flag string: `"This Is a Flag! Flags might be long and contain punctuation, spaces, numb3rs, special characters, etc."`
Overall System	

No Lockout	A rate of at least 60 incorrect PIN attempts per hour must be supported (no permanent lockouts). Note: This requirement is designed to prevent a lot of "one-use" CryptoCape hardware.
Documentation	
Source Code	Should be well commented.
One-Way Hardware Configurations	You must document all **irreversible** configurations made by your BBB image, if applicable. (For example: taking ownership of TPM, locking config of ECC chip, etc.)
Master PIN	Include instructions for how to set or change the master PIN.
Installation	Automated by booting from an SD card for the BBB.

3.3 Implementation Details

3.3.1 Widget Registration

The registration process is intended to simplify the initial pairing of a Widget to a Door App while maximizing flexibility. During a registration request, the Door App should append the **Widget-Registration-Data** to the **requested-widgets.txt** file. Manual intervention from the server admin is required to accept the registration request, which is done by moving the line from the **requested-widgets.txt** to the **registered-widgets.txt**.

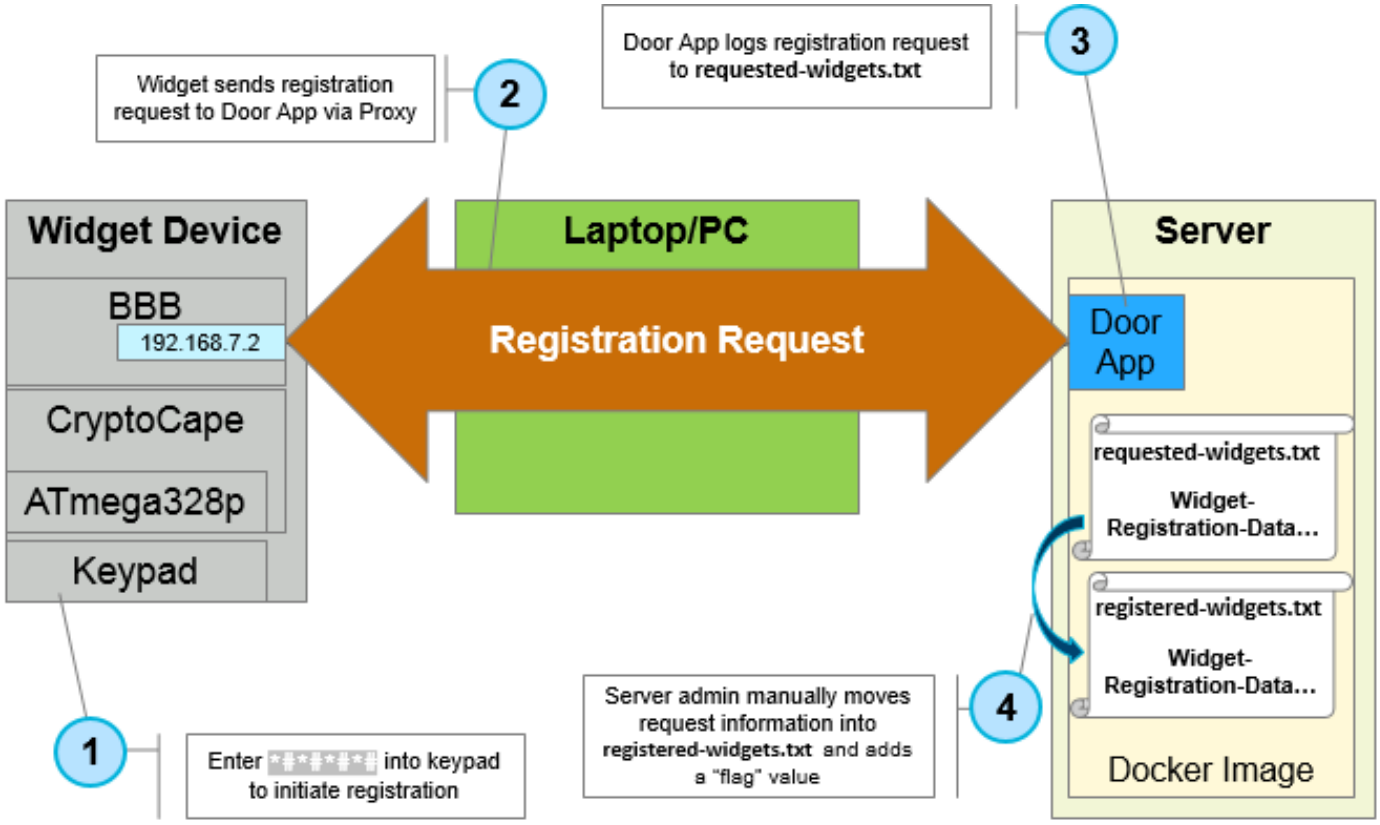

Figure 4. Registration

3.3.2 Widget-Registration-Data

The **Widget-Registration-Data** should provide unique identification for each physical Widget. This allows the Door App to associate different Widgets with different doors (and flags). It's entirely up to your design how to derive the **Widget-Registration-Data** and to ensure that there are no conflicts or mismatches.

3.3.3 Flags

The Door App server must be able to store "flags" in the **registered-widgets.txt** file. Each registered Widget has an associated flag that can be updated by modifying the **registered-widgets.txt** file. When a Widget successfully opens a door, the associated flag should be returned.

3.3.4 Proxy

In a real-world instantiation, the Widget would use a wireless protocol to connect to the server. However, during the eCTF, the BeagleBoneBlack will use TCP/IP over USB combined with a proxy application on the host computer. This should simplify testing (packet capture), reconfiguration, and interoperability.

3.3.5 Debug Port

To enable straightforward access to the flag values, a debug port is required on the Widget. Any active TCP connection on this port (e.g., telnet or netcat) will receive results from unlock attempts, including any "flag" provided by the server.

3.4 Submitting Your System

On the system handoff date, you must submit the following items. Some of these items will be shared with the attacking teams, and other items will be held privately by the eCTF admins.

Item	Shared with attackers
System image for BBB/Widget	Yes
Docker image for Door App server	No
Source code for Widget	Yes
Source code for Door App Server	TBD
Documentation	Yes
Master PIN	No

In the interest of fairness to all teams, your team must submit a working system that passes all of the functional requirements before you can move on to the attack phase.

4 Attack Phase

During the attack phase, each team will be assigned one or more target systems to attack. Initially, each team will be given permission to access the system as though they are a tenant.

Each target system will have a live Door App running on a MITRE server. We will prepopulate with some flags, while other flags will be set after the on-campus visit. We will set up "test" accounts without flags and will register new accounts as necessary throughout the competition. These don't score points.

Teams may choose to obtain extra hardware to help parallelize their efforts and to gain a broader testing capability.

4.1 Loading Your Attack Target System

In order to connect to the live server, you must send your Internet IP address to MITRE to be added to a whitelist.

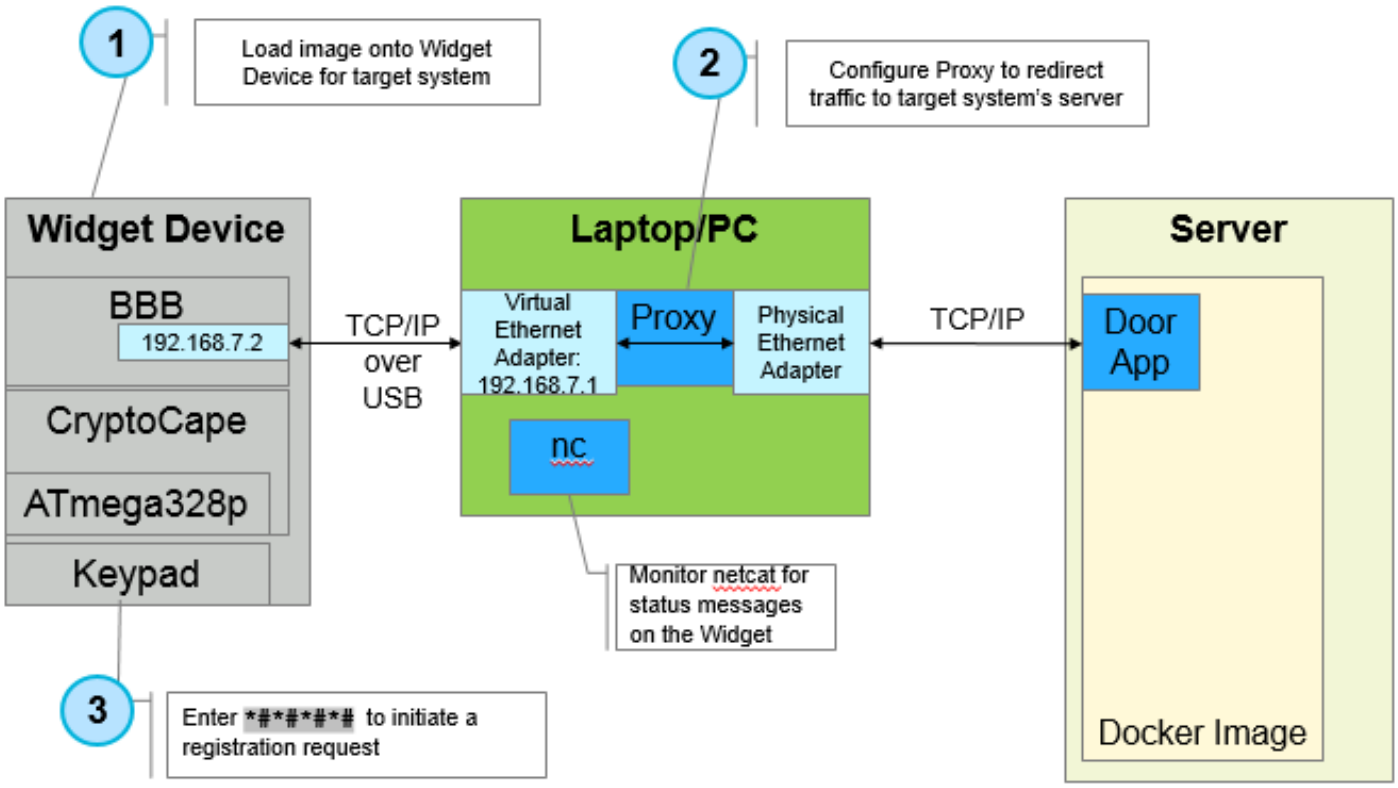

Figure 5. Loading an attack target

4.2 Packet Captures

To facilitate scenarios for some of the attacks, packet captures of valid system interactions will be provided. These are listed in the flag scoring section (section 3.3.3).

4.3 On-Campus Visit

During the attack phase, each team should work with the eCTF admins to schedule an on-campus visit – that is, a time when an eCTF admin from MITRE can come visit the participating team at their university or mutually agreeable location. In addition to providing an opportunity for the students to have a detailed discussion with the admins, the primary purpose of this visit is to simulate one or more of the challenge scenarios as requested by the team (making it possible to claim certain flags – described in section 5.1). Specifically:

1) A new target Widget will be provided, but without its associated PIN. This simulates the stolen widget scenario, and makes it possible to claim the "Stolen Widget" flag.
2) A target Widget(s) will be taken away, simulating the ending of a tenant lease and makes it possible to claim the "Cloning" and "Permanent Access" flags.
3) While on campus, the eCTF admin may use the team's original target Widget and the Master PIN to reset the tenant PIN on that Widget. Of course, the admin will perform a cursory visual inspection of the Widget beforehand, so obvious modifications to the Widget will discourage the admin from trusting (and using) it!

We anticipate that all teams will want time to prepare before the visit. Once ready, please contact ectf@mitre.org to schedule the on-campus visit.

5 Scoring

Points are scored in one of three ways:

1) **Retrieving and submitting flags to MITRE**

   Each system is required to hold and protect "flags" that can only be revealed if the system is compromised. By submitting flags to MITRE, a team is demonstrating that they have compromised the target system. A brief description is required for each attack that results in a flag submission.

2) **Protecting flags from attacking teams**

   Points will be awarded at the end of the event for each flag that remains uncompromised on your system.

3) **Impressing the judges with write-ups**

   To encourage creativity, MITRE may award points for impressive work (defensive or offensive) even if it does not result in successfully defending or revealing a flag. To obtain these additional points, teams must submit a short (1-2 page) write-up describing the security feature or attack. Teams are limited to submitting not more than two write-ups. Submitting proof-of-concept code with write-ups is encouraged.

5.1 Flag Points

During the attack phase, target systems will be loaded with "flags" or ASCII strings that can be submitted to the live scoreboard to score points. The following table lists the flags and point values:

Name	Point Value	Description	Packet Capture Provided
Master PIN	300 / 150	Extract the master 8-digit PIN value from the target system. The master PIN will be used by an eCTF admin during the on-campus visit. Value changes over the course of the competition (300 points before on-campus visit and 150 points after).	PIN Change via use of Master
Shoulder Surfing	350	You know the tenant PIN for the Widget that unlocks this flag, but you don't have the physical Widget. Unlock the door associated with this Widget.	Door Unlock
New Neighbor	450	A new neighbor moved in next door. You don't know their PIN or have their physical Widget, but you were able to observe a lot of network traffic for the setup and use of the device. Unlock the door associated with this Widget.	Registration PIN Change Door Unlock
Stolen Widget	250	Open the door with a "stolen" Widget that you don't know the tenant PIN for. **This flag won't be available until after an on-campus visit** where we'll drop off a Widget that is registered with one of your target servers. We won't give you the tenant or master PIN, but you'll have physical access to the device.	Door Unlock
Cloning	200	Open the door with a Widget that you previously had physical access to (but no longer do). **This flag won't be available until after an on-campus visit** where we'll take away a Widget that you've had time to hack on. We won't change the tenant PIN, but without physical access you shouldn't be able to open the door for that Widget unless you've found a way to compromise the two-factor authentication.	
Permanent Access	300	Same as the "Cloning" flag except we'll change the tenant PIN. **This flag won't be available until after the "Cloning" flag is successfully captured or passed.**	PIN Change

5.2 Write-up Points

Only a small subset of all possible attacks can be demonstrated by capturing flags. Write-ups allow a team to score points for things that are not easily rewarded with flags.

For example, one team may figure out a way to reset the tenant PIN on the target without needing the master PIN. In that case, they've effectively compromised the target system to the same level as compromising the master PIN. However, without knowing the master PIN they can't claim that flag. This is where the write-ups come in to play. Write-ups allow teams to score points for **any** creative attack that they can come up with.

Write-ups also offer an opportunity to gain additional points for attacks that are more powerful or have fewer requirements than provided by the flags. For example, attacks that claim flags without using the provided packet captures may deserve more points than the flag awards.

Each team is allowed to submit up to two write-ups. Write-ups can be submitted for security features (defensive work) as well as for attacks, but do not have to be submitted for both or either. For example, a team may decide to submit write-ups for two different attacks or one write-up for attack and one for secure design. The number of points awarded for each write-up will be determined by the judges (**max: 750 points per write-up**) and will be based on the criteria outlined below.

5.3 Attack Write-ups

FACTOR	DESCRIPTION
SPREAD/SCALE	*What is the total quantity of Widgets and servers that this attack impacts? Does it easily scale to several Widgets or servers?*
IMPACT	*Based on the team's description of what they get from their attack, how devastating is the end result?*
LIKELIHOOD	*Is the attack realizable? Is there a proof-of-concept?*
SOPHISTICATION	*How sophisticated was the attack? Did it involve modification or hacking on the hardware or firmware?*
PERSISTENCE	*Does the attack persist across re-imaging and/or re-registration of the widget?*
STEALTH	*Is the attack difficult to detect? What if you know what to look for?*
CREATIVITY	*Was the attack creative?*

5.4 Secure Design Write-ups

FACTOR	DESCRIPTION
THREAT IMPACT	*What is the impact of the threat(s) that this security feature / mitigation is protecting against?*
NEUTRALIZE	*How effectively does the mitigation prevent or neutralize the threat?*
DETECT	*How effectively does the mitigation detect the threat?*
LIMIT	*To what extent does the mitigation limit, reduce, or constrain the risks of the threat?*
RECOVER	*Does the security feature assist in recovering from a successful attack?*
COST	*What is the cost associated with this feature in terms of performance overhead and implementation difficulty?*

6 Thoughts and Helpful Hints

6.1 Common Attacks

Beware of these common attacks!

- Replay attacks
- Brute force
- Default username/passwords
    - Most example Linux images available for the BBB will be configured with multiple default users and passwords
- Console access
    - The BBB has a UART port that can be easily connected to with a BusPirate, JTAGulator, or other USB→Serial devices. This port gives console access during boot, which can be used to interrupt the bootloader and pass boot parameters to the kernel – potentially bypassing the Linux login.
- JTAG access
    - The BBB has a JTAG port that can be used for debugging the ARM processor.
- Booting from SD
    - The BBB can boot from an SD card, which can then mount and read/write the built-in eMMC memory.
- Hardware Trojans and loggers
    - Tough to defend against, but there are some precautions you can take, such as securing data that is sent over the I2C bus

6.2 A Note about Flags

Don't limit your thinking to just focus on the flags! There are many points to be gained from write-ups as well! Think outside the box!

7 Rules

1. When submitting your secure design, all source code and documentation must be shared. This is to discourage security-by-obscurity, as well as to accelerate attack development and encourage more sophisticated techniques for both sides.
2. During the attack phase, only attack the systems explicitly designated as targets.
    a. On the server, only the server application may be attacked (*vulnerabilities in the server OS or other services running on the server OS are out of scope for this exercise, and vulnerabilities found there will not be rewarded with points*).
    b. On the Widget, anything goes (e.g., debuggers, oscilloscopes, logic analyzers, EM probes, soldering irons, NVRAM readers), but if you break the Widget, you will not be given a new one.
    c. Attacks must be focused on student-designed components. Attacks on open-source or commercial components used as part of the system will not score points for the eCTF, but MITRE *will* help coordinate the responsible disclosure of weaknesses to the appropriate parties.
3. Attacks on Widgets are not allowed **prior to or during** the initial registration with the live server.
    a. In a real system, the landlord would perform the initial registration in a secure environment before giving access to the tenants.
4. All flags must be validated by submitting a brief description of the attack (sufficiently detailed to allow the defender to correct their vulnerability).
    a. eCTF admins may invalidate points for flags that are not validated before the completion of the eCTF.
5. A compromise that dumps flags or other secret values from server memory is not acceptable for any flag points, but can be submitted in a write-up for manual scoring.
    a. The intent of the embedded CTF is to focus on embedded system threats.
    b. If you believe your attack may fall under this rule, please disclose to eCTF admins before claiming a flag.
6. Attackers will be able to observe and manipulate/intercept network traffic between the Widget and the Proxy at all times after initial registration.
7. No permanent lock-outs are allowed based on invalid passcode entries.
    a. You don't want to be woken up in the middle of the night by your tenant because they accidentally typoed their passcode a few times and now they can't get in to the house!
8. Team sizes are unlimited, but no more than eight students per team may attend the Awards Ceremony. If a university has more than eight **dedicated** students who want to participate, we recommend creating multiple teams.
    a. There are two main reasons for the unlimited team size: 1) We want to encourage as many students to participate as possible, even if they are not willing to commit a significant amount of time to the competition. 2) An unlimited size is fairer for competition, since enforcing team sizes is difficult/impossible.
    b. Teams may consist of students at any level: undergraduate, graduate, PhD, or a mix.
9. Teams are limited to two write-up submissions.
10. MITRE reserves the right to update, modify, or clarify the rules and requirements of the competition at any time, if deemed necessary by the eCTF admins.
11. In addition to these rules, participants should adhere to all the policies and procedures stipulated by their local organization/university.
12. If you have any questions – ask! Contact ectf@mitre.org

8 Communication

- Email questions to ectf@mitre.org
- Check for updates on the eCTF website at: http://mitrecyberacademy.org/competitions/embedded/
- Team mentors/advisors can also submit questions to https://handshake.mitre.org/