

# Teaching CyBOK Through Cyber Physical Systems

## Future Funfair

**UWE  
Bristol** | University  
of the  
West of  
England

Funded By



## Contents

Teaching CyBOK Through Cyber Physical Systems.....	0
Introduction: .....	2
Setup:.....	2
Components.....	2
Wireless Access Point .....	3
Pi Build Hat.....	3
Student Laptops .....	4
Staff Pi 400 Attack machine .....	6
Attack descriptions.....	7
Denial of Service.....	7
Code Injection .....	7
Man in the Middle .....	7
Stop Rides .....	7
Flush.....	7
Software Installation .....	8
Software Images.....	8
Document Revision History.....	9

## Future Funfair

### Introduction:

This exercise offers students an engaging introduction to the field of cyber security, employing a fun and interactive setting: a Lego Funfair. The primary focus is the exploration of Cyber Physical Systems, in this instance, the Lego Funfair, and their vulnerability to various forms of cyber-attacks.

During this activity, students will delve into three specific forms of attack, namely: Denial of Service (DoS), Code Injection, and Man in the Middle (MitM) attacks. Each of these will be targeted at the LEGO motors, which animate the fairground simulations, or a LED colour matrix. The manifestation of these attacks will range from altering the motors' speeds to changing the LED matrix's colours.

Students will utilise Wireshark to monitor network activity and analyse network traffic. The aim is to identify regular traffic patterns and comprehend the distinguishing characteristics of the individual attacks as they occur. Armed with this knowledge, students will then be able to develop and implement strategies to mitigate these attacks.

### Setup:

#### Components

Each set of systems, capable of accommodating two groups of five to six students, consists of the following components:

- 2 Lego funfair rides using one motor each
- 1 colour matrix
- 1 Raspberry Pi with Pi Build Hat acting as the control server.
- 1 Pi 400 (complete with screen) for staff to orchestrate attacks from (Note: Any machine with an SSH Client can be used for this purpose)
- 2 student laptops equipped with Kali Linux.

1 Raspberry Pi acting as a Wi-Fi access point will provide connectivity for all systems, this will need to be a Raspberry Pi model 3 for the provided disk image to work. The control server code allows for three sets of the above systems to operate at the same time, allowing for a total of 30-36 students.

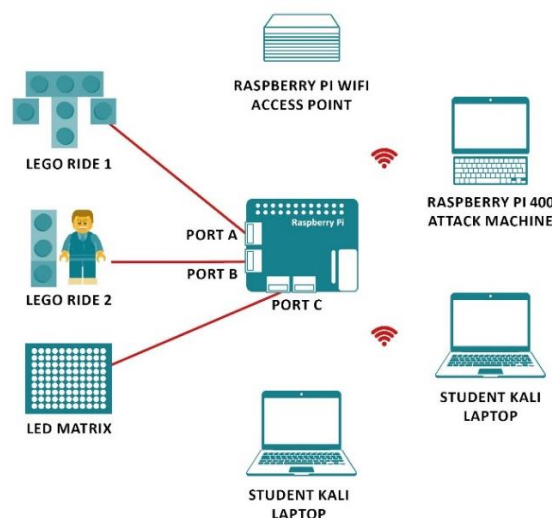


Figure 1: System Setup

### Wireless Access Point

The wireless access point configures a local Wi-Fi network to which all wireless components can connect. Student laptops and the Pi 400 should connect to this network. The Pi Build Hat device has been pre-configured to connect to this network.

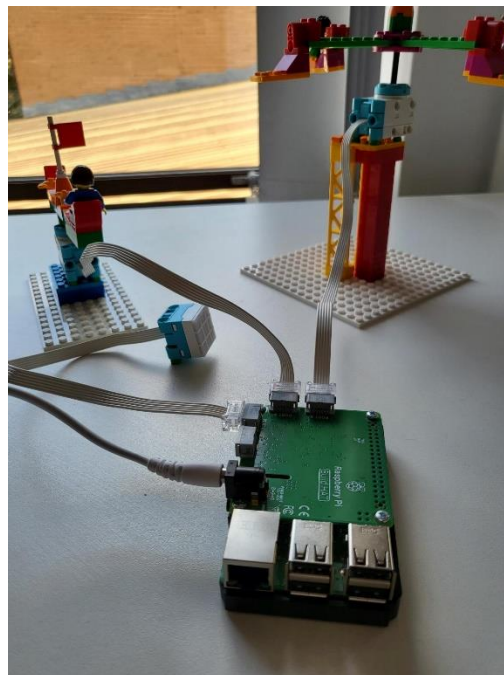
*Table 1: Wi-Fi configuration details*

<b>SSID:</b>	<b>UWEcyber_RasPwnOS</b>
<b>Password:</b>	<b>UWEcyber_In53cur3!</b>

### Pi Build Hat

It is **essential** that the Lego components be plugged into the Pi Build Hat **before** the Pi Server is powered on.

The Lego motors **must** be plugged into ports **A** and **B** on the Pi Build Hat. The LED matrix **must** be plugged into port **C**.



*Figure 2: Ride connections*

By default, we have the capacity to support up to three sets of kit on a network (equating to 6 rides) using three separate Pi Build Hat servers. The server image supplied will automatically configure the Pi to connect to the local Wi-Fi network and assign itself the host IP address 192.168.99.101.

Since each host necessitates a unique IP address, you can execute the 'configure\_ip.sh' script. By supplying the kit number (1, 2 or 3) as a parameter to the script, it will automatically adjust the host IP address to 192.168.99.101, 192.168.99.102, or 192.168.99.103 respectively. After a reboot the server will boot with the allocated IP address.

## Student Laptops

Students will utilise the Kali-equipped laptops to investigate and interact with the system. Wireshark must be operational on the student laptops, monitoring the wireless interface. This is essential for the students to analyse network traffic and identify the nature of the attack(s). To further simplify the traffic seen by the students, a Wireshark display filter of “udp && !dns” can be used.

The student User Interface (UI) is accessed directly from the laptop. This UI not only offers additional information about the attacks but also provides a platform for students to implement suitable countermeasures.

The UI will be pre-loaded and executed via a local Python HTTP server. To initiate the server, navigate to the UI directory and execute the following command:

```
$ python -m http.server
```

Access to the UI is available through a web browser at the following address: <http://localhost:8000>.

Example screenshots are shown in Figure 3 and Figure 4

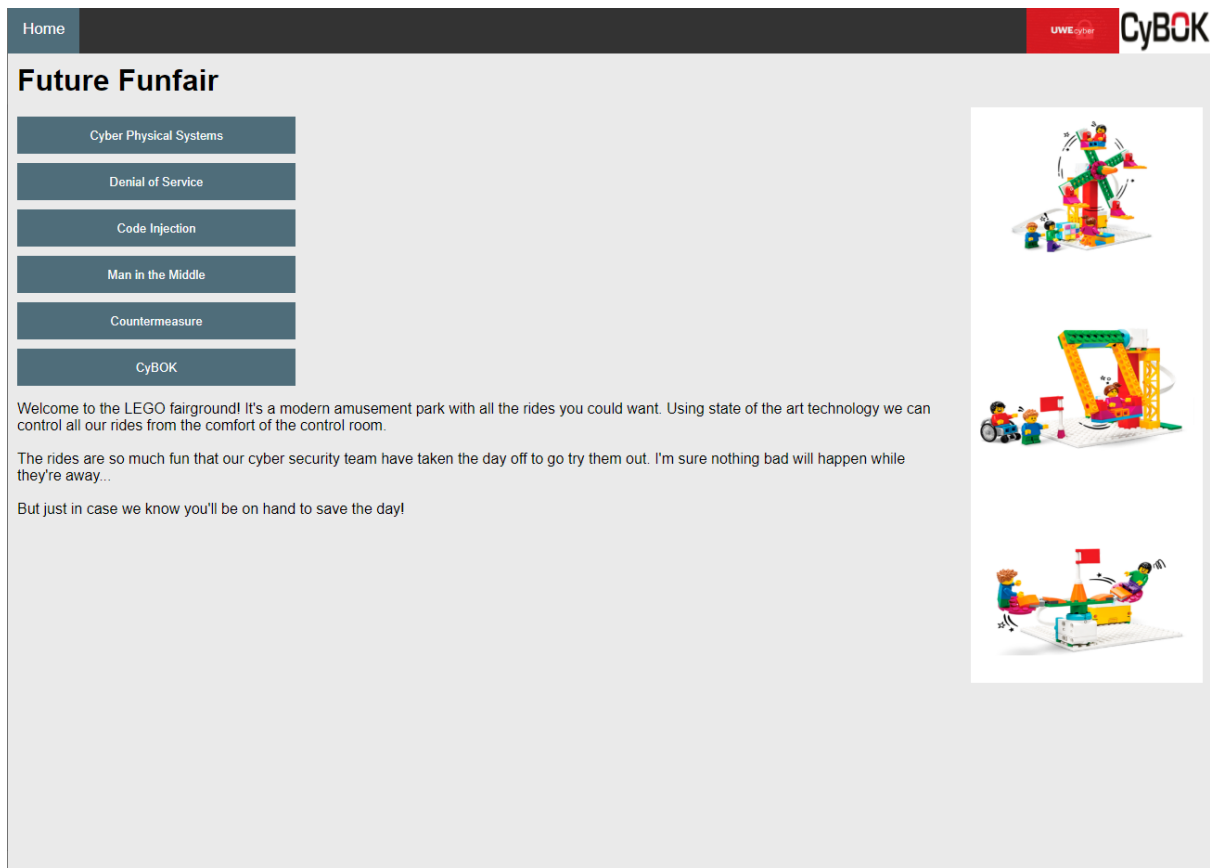


Figure 3: UI Main Menu (Splash page)

## Countermeasures

Oh are we glad you're here! The rides are under attack and to make things worse our cyber security team are on them!

We need your help to save the day (and the cyber security team). Thankfully we've been learning all about how to protect Cyber Physical Systems from the [Cyber Security Body Of Knowledge \(CyBOK\)](#). We've put the important bits below to help you

The different attacks can all be stopped in different ways. Click below to learn more and stop the attackers

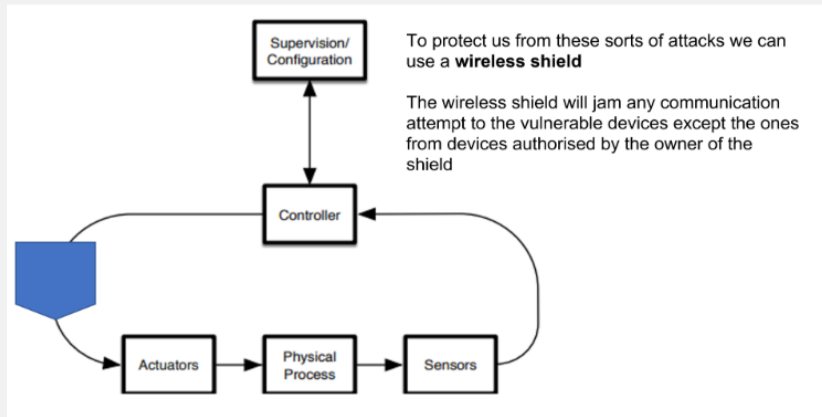
Don't forget we need to know the IP address of our Pi Server - Otherwise our commands won't go anywhere! Add it in below

The IP address of our Pi Server is:

### Denial of Service

To counter a DoS attack we need to stop the sensor being flooded with packets.

One way we can do that is by using a wireless shield. This will block all unexpected traffic.



Before we can enable our shield though we need to be able to identify which traffic is the DoS and which is the normal traffic.

Enter the IP address of the attacker below. Once you get it right the shield will be enabled.

The IP address of the attacker is:

Figure 4: UI - Countermeasure page (DoS section)

### Staff Pi 400 Attack machine

Staff members will employ the Pi 400 to remotely access the student laptops and instigate the attacks directly from the laptop. By initiating attacks directly from the student laptop, this simplifies the traffic seen by the students in Wireshark to that from the local machine only. Each Pi 400 will support 2 student laptops.

A terminal session shall be used to SSH to the student laptop and initialise the attack script. The Lego kit is identified as either Kit A or B, depending on whether it's connected to Motor Port A or B. Upon initiating the attack script from the SSH session, the script will confirm which kit you wish to control for this session.

The attack scripts operate via interactive CLI prompts, enabling staff to start IDLE traffic, initiate an attack, halt an attack, and so on. While the attacks are ongoing, students should be prompted to explore the traffic and input the relevant mitigation via the UI on their laptop. Once the mitigation is activated, staff can advance the script and terminate the attack. Follow the onscreen prompts to select your attack.

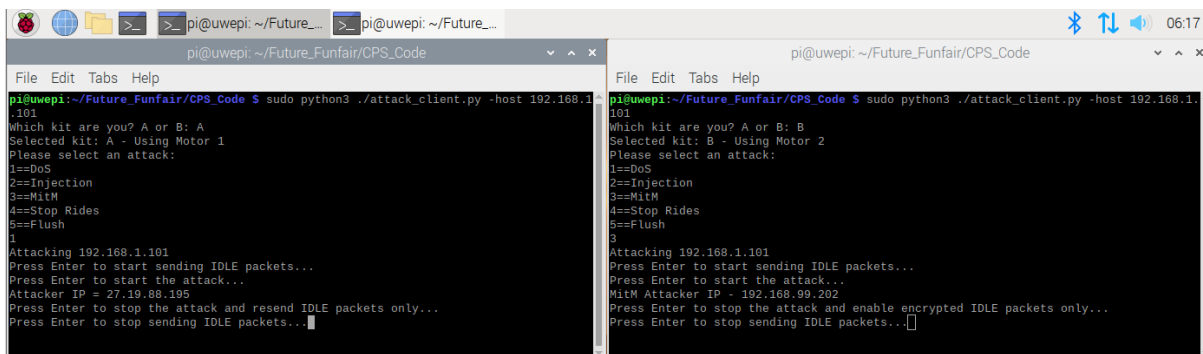


Figure 5: Attack scripts controlling different rides

The Man in the Middle attack targets a LED matrix (of which there is only one in this setup). **ENSURE SURE YOU COORDINATE WITH THE OTHER GROUP SO ONLY ONE MITM ATTACK IS RUN AT A TIME.** If both groups try to run the MitM attack at the same time this will not work as expected.

All the machines are running on a local WiFi network. There is no internet connectivity. Please do not try and google anything using the laptops. This won't work.

## Attack descriptions

### Denial of Service

This form of attack interrupts the motors of the target ride, thereby halting its rotation. The target system is overwhelmed by a flood of junk packets. As the students monitor the Wireshark while the ride operates normally and then during the attack, they should be able to discern the difference. To combat this, students need to locate the attacker's IP and activate the "Wireless Shield" via the UI, which will block the attacker. Once the attacker has been successfully blocked, let the attack proceed for a short while so the students can observe that the attack traffic is still being sent but no longer has any impact.

### Code Injection

This attack affects the motors of the target ride causing the ride to spin faster. While students monitor Wireshark during the normal operation of the ride and then during the attack, they should be able to identify the difference in network traffic. During the attack packets will be sent from the attacker instructing the ride to run at the 'fastest' speed - this is the code injection. To counter this attack, students need to identify the attacker's IP address and "Enable the Firewall" via the UI, which will block the attacker. Once the attacker has successfully been blocked, let the attack continue for a short while so that the students can see the attack traffic is still being sent but is no longer having any impact.

### Man in the Middle

MitM attack will target the LED matrix, changing its colours. This will be easily identifiable due to the change in the system; it will also cause the target ride to start spinning faster. As the students monitor Wireshark both during the normal operation, and then during the attack, they should observe a difference in network traffic. The traffic, instead of being destined to the Pi server, is being "intercepted" and the command "green" is being changed to "red" by the attacker. To mitigate the issues, students need to locate the attacker's IP address and activate the "Bump-in-the-Wire device" via the UI. The device will block the attacker and enable encryption.

Once students have correctly blocked the attacker return to the attacker's raspberry Pi and press Enter (a popup will occur which tells them / you to enable encryption). This will start sending encrypted traffic. Let this run for a little bit so they can see that the traffic is now encrypted (and therefore our attacker can't read or intercept it).

### Stop Rides

If something goes awry or the script crashes, there may be a need to halt the rides and restart. Option 4 on the attacker script will do this.

### Flush

As we employ IPTables on the Pi server to block our attacker IP addresses (and these are all statically set) it is necessary to flush the iptables rules after each complete run of the activity (not after each individual attack). This ensures the attacks function as expected for the next group. **Please ensure this is done between groups of participants; otherwise, the attacks may not perform as expected for the subsequent group.**



## Software Installation

All software images and support material can be found at: <https://go.uwe.ac.uk/uwecyber> under the **Resources** sub-section.

The attack and server software can be independently downloaded from GitHub - [https://github.com/amills157/Future\\_Funfair](https://github.com/amills157/Future_Funfair)

### Software Images

For ease of installation, we provide three software images that have been pre-installed with software.

Image Name	Description
UWEcyber-RasPwnOS-1.0.1.iso (8GB SD card)	Image for the <b>Wireless Access Point</b> . Provides the local <b>UWEcyber_RasPwnOS</b> Wi-Fi network  <b>Login:</b> pi <b>Password:</b> pwnme!
lego_cps_server.img (8GB SD card)	Image for the <b>Raspberry Pi Build Hat Server</b> . Pre-installed with the server script, set to autorun on initial boot.  <b>Login:</b> uwecyber <b>Password:</b> uw3cyb3r
UWEcyber-KaliPi-1.0.1.iso (32GB SD card)	Image for the Pi 400 attack machine based on Kali Linux. (Note: Any image that includes an SSH client could be used for this machine)  <b>Login:</b> kali <b>Password:</b> kali

These images can be burnt to the Raspberry Pi SD card using an imaging tool such as [Balena Etcher](#).

The attack software and UI will need to be individually installed on the student laptops from the Github repository. The student laptops will need to be connected to the **UWEcyber\_RasPwnOS** Wi-Fi network.

## Document Revision History

<b>Version</b>	<b>Date</b>	<b>Description</b>
v1.0	28 <sup>th</sup> June 2023	Initial Release