WORKSHEET

F11.9 Project: secure the robot

Cyber security · GCSE · OCR J277 1.4.2, AQA 8525 3.6.3, Edexcel 1CP2 4.2.1 · about 25 min

BugBotLab

Name _____ Class _____ Date _____

What this lesson is about

Authenticate with a password, decrypt an encrypted command, and act on it.

- The protocol
- Try the pieces
- Plan it first

Questions

5 marks in all

1. Why does the robot authenticate before taking a command? [1 mark]

- ☐ **A** So only someone with the password can control it
- ☐ **B** To make it faster
- ☐ **C** To save battery
- ☐ **D** To encrypt the mat

2. Why is the command encrypted? [1 mark]

- ☐ **A** So a listener on the radio cannot understand it
- ☐ **B** To make it shorter
- ☐ **C** So it arrives faster
- ☐ **D** So the robot can ignore it

3. A listener hears the Base's encrypted reply, KTWBFWI 25. What can they read without the key? [1 mark]

- ☐ **A** Only the number 25, because the Caesar cipher shifts letters and leaves digits alone
- ☐ **B** The whole command
- ☐ **C** Nothing at all
- ☐ **D** The key

4. Authentication and encryption together protect against what? [1 mark]

- ☐ **A** Impostors giving commands and listeners reading them
- ☐ **B** Only power cuts
- ☐ **C** Only malware
- ☐ **D** Only slow networks

5. If the wrong password is sent, what should the robot do? [1 mark]

- ☐ **A** Not move
- ☐ **B** Drive anyway
- ☐ **C** Send its password
- ☐ **D** Turn off the radio

The task: secure the robot

Authenticate with the Base using the password `bugbot42`, decrypt its reply with a Caesar cipher (key 5), and carry out the `FORWARD <cm>` command to reach the green depot, without collisions. Do not send any command before the password, and do not type the distance: read it from the decrypted message. The `caesar` helper is provided.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def caesar(text, key):
    out = ""
    for ch in text:
        if ch.isalpha():
            base = ord("A")
            out = out + chr((ord(ch.upper()) - base + key) % 26 + base)
        else:
            out = out + ch
    return out

def ask(question, seconds=1.5):
    send(question)
    for tick in range(int(seconds * 10)):
        wait(0.1)
        for sender, text in messages():
            return text
    return None
```

Plan your program here, then type it in and press Run.

Do it on the robot

www.bugbotlab.com/learn/f11-9-project-secure-the-robot/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Try sending the wrong password first. Does your program correctly refuse to move?
2. Someone else on the mat is listening. The Base's reply is `KTWBFWI 25`: what can they read without the key? (Look at what the Caesar cipher leaves alone.) And what did they learn when your robot sent the password?
3. Encrypt your own reply to the Base to confirm you have arrived, and send it.