



## F11.1 Why cyber security

Cyber security · GCSE · OCR J277 1.4.1, AQA 8525 3.6.1, Edexcel 1CP2  
5.3.1 · about 15 min

### What this lesson is about

Confidentiality, integrity and availability, who attacks and why, and ranking risk.

- What are we protecting?
- Who attacks systems, and why?
- The weakest link is usually people
- A quick risk model

### Questions

5 marks in all

1. What does confidentiality mean in security? [1 mark]

- ☐ A Only the right people can read the data
- ☐ B The data is always available
- ☐ C The data is never changed
- ☐ D The data is backed up

**Answer:** A. Confidentiality is about keeping data secret from those who should not see it.

2. Flooding a website until it crashes attacks which goal? [1 mark]

- ☐ A Availability
- ☐ B Confidentiality
- ☐ C Integrity
- ☐ D Authenticity

**Answer:** A. Availability is whether the system works when needed.

3. What is often the weakest link in security? [1 mark]

- ☐ A People
- ☐ B Firewalls
- ☐ C Encryption
- ☐ D Cables

**Answer:** A. Many breaches begin by tricking a person.

4. Why might an insider be a threat? [1 mark]

- ☐ A They already have access they can misuse
- ☐ B They cannot log in
- ☐ C They have no motive
- ☐ D They work from home

**Answer:** A. An unhappy employee can misuse access they were given.

5. A risk has likelihood 4 and impact 5. What is its risk score, if score is likelihood times impact?

[1 mark]

**Answer:** 20.  $4 \times 5$ .

## The task: rank the risks

Give each risk in `risks` a **score** of likelihood times impact. Print each as `<name>: <score>`, sorted from the highest score to the lowest. Then print `biggest risk: <name>` for the one at the top.

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

# name, likelihood (1 to 5), impact (1 to 5)
risks = [("weak passwords", 4, 5), ("stolen laptop", 2, 4), ("phishing email", 5, 4),
("power cut", 3, 2), ("out-of-date software", 4, 3)]
```

**The hint students can ask for:** Give each risk a score of likelihood times impact, then sort the list by that score, highest first. The biggest risk is whatever ends up at the front.

### A solution

```
from bugbot import *
connect()
risks = [("weak passwords", 4, 5), ("stolen laptop", 2, 4), ("phishing email", 5, 4),
("power cut", 3, 2), ("out-of-date software", 4, 3)]
risks.sort(key=lambda r: r[1] * r[2], reverse=True)
for name, likelihood, impact in risks:
    print(f"{name}: {likelihood * impact}")
print("biggest risk:", risks[0][0])
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.