



F11.7 Defending a network

Cyber security · GCSE · OCR J277 1.4.2, AQA 8525 3.6.3, Edexcel 1CP2
4.2.1 · about 15 min

What this lesson is about

Firewalls, anti-malware, access levels, updates, backups, penetration testing and policies.

- The defences
- Penetration testing
- Network policies
- How a firewall decides

Questions

5 marks in all

1. What does a firewall do? [1 mark]

- ☐ A Blocks network traffic that its rules do not allow
- ☐ B Removes viruses
- ☐ C Encrypts files
- ☐ D Backs up data

Answer: A. It controls what traffic goes in and out.

2. What is penetration testing? [1 mark]

- ☐ A Attacking your own system on purpose to find weaknesses
- ☐ B Installing anti-malware
- ☐ C Encrypting a network
- ☐ D Training staff

Answer: A. You fix the holes before a real attacker finds them.

3. Why are user access levels useful? [1 mark]

- ☐ A A breached account can only reach a limited amount
- ☐ B They make passwords longer
- ☐ C They encrypt files
- ☐ D They speed up the network

Answer: A. People and programs get only the access they need.

4. Why do automatic updates help security? [1 mark]

- ☐ A They fix known security holes quickly
- ☐ B They make the computer faster
- ☐ C They add more storage
- ☐ D They remove old files

Answer: A. Most attacks use holes that a patch already fixed.

5. What is a network policy?

[1 mark]

- ☐ A The rules an organisation sets for staying secure
- ☐ B A kind of firewall
- ☐ C A password
- ☐ D A backup drive

Answer: A. A weak or ignored policy is itself a vulnerability.

The task: a firewall

Complete the firewall. For each packet in `packets` (an address and a port), check the `rules` in order and take the action of the first rule that matches. A rule matches if its port is `any` or equals the packet's port, and its address is `any` or equals the packet's address. Print `<address>:<port> -> allow` or `-> block`, and at the end print `blocked <n> of <total>`.

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

# each rule: (action, address, port). First match wins.
rules = [
    ("allow", "10.0.0.9", 22),
    ("block", "any", 22),
    ("allow", "any", 443),
    ("allow", "any", 80),
    ("block", "any", "any"),
]

# each packet: (address, port)
packets = [("10.0.0.9", 22), ("10.0.0.5", 22), ("10.0.0.5", 443), ("10.0.0.2", 8080),
            ("10.0.0.5", 80)]
```

The hint students can ask for: For each packet, walk the rules in order and stop at the first one that matches, taking its decision. A rule matches when its address and port either match the packet or are the wildcard. Count how many you blocked.

A solution

```
from bugbot import *
connect()
rules = [
    ("allow", "10.0.0.9", 22),
    ("block", "any", 22),
    ("allow", "any", 443),
    ("allow", "any", 80),
    ("block", "any", "any"),
]

packets = [("10.0.0.9", 22), ("10.0.0.5", 22), ("10.0.0.5", 443), ("10.0.0.2", 8080),
            ("10.0.0.5", 80)]
blocked = 0
for address, port in packets:
    decision = "block"
    for action, rule_addr, rule_port in rules:
        if (rule_port == "any" or rule_port == port) and (rule_addr == "any" or rule_addr
        == address):
            decision = action
            break
    if decision == "block":
        blocked = blocked + 1
    print(f"{address}:{port} -> {decision}")
print(f"blocked {blocked} of {len(packets)}")
```

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

