Solutions

Each answer shows the steps, then the **result**.

2.1

- an adversary who gains access to the device or the drive can read any unencrypted file directly; physical security can fail - a stolen laptop, a disposed disk, a remote compromise - and encryption is what makes that access useless

2.2

- any two of: change system settings; access any file or application on the system; install software; create or alter accounts. It matters because compromising one administrative account gives the adversary all of that at once, whereas a standard account is limited to what that user could already reach

2.3

(a)

- directory traversal

(b)

- SQL injection

(c)

- cross-site scripting

(d)

- buffer overflow

2.4

- SQL commands and control characters typed into an input field become part of the query the application runs, so the database returns records the user was never authorised to see - unauthorised access to sensitive data, which is a breach of confidentiality

2.5

- the malicious JavaScript is stored in or reflected by the website, and the victim's browser executes it because it arrived from a site the browser trusts, not from the attacker

2.6

- a buffer is a designated section of memory with a **fixed size** where user input is written; if more data is supplied than was allotted, it runs past that section into adjacent memory, which can crash the system or cause it to execute code outside the program's security policy

2.7

(a)

- it checks the **length** of the input and rejects anything longer than the buffer can hold

(b)

- it checks the characters, rejecting or escaping the control characters and SQL keywords so the input is treated as data rather than as commands

2.8

(a)

- confidentiality, and integrity if it can be edited; **high** - the data is governed by law and the weak access control makes exploitation highly likely

(b)

- confidentiality; **low** - a short key is a real weakness, but a lunch menu is not sensitive, so the impact is minimal

(c)

- confidentiality; **moderate** - the data is sensitive but is encrypted, just not strongly enough

3.1 C

- Modifying a URL or GET request to reach the server's file system is directory traversal

3.2 B

- Validation is what stops input being treated as instructions

3.3 B

- A is SQL injection, C is directory traversal, D is XSS

3.4 B

- A is high; C and D are low

3.5

(a)

- **directory traversal**. The `../` sequences climb out of the orders directory to reach a database configuration file, which typically contains the credentials for the database itself - so one request could yield full access to all customer data

(b)

- **cross-site scripting**. The harm falls on **other customers**, not on the bookshop's server: the script is stored in the review and executes in each later visitor's browser, where it can steal session cookies, capture what they type, or redirect them

(c)

- **SQL injection**. Input passed straight into a query means typed control characters become part of the query, risking **confidentiality** if it returns other customers' orders and **integrity** if it alters or deletes records; availability is at risk too if the injected command drops a table

(d)

- every compromised staff account is an **administrative** account, so an adversary who phishes any one employee can change system settings, reach every file and install software - the attacks above stop being limited to what one user could see and become full control of the shop's data, which is a direct failure of least privilege

(e)

- Four changes, each on a different vulnerability: validate and parameterise the search input so it can never be executed as SQL; escape or sanitise review content before display, so a script tag is shown as text; encrypt the order files **and** move them out of the web-served directory; and give staff standard accounts, reserving administrative rights for the few who need them. Rating: **high** - the data is customer orders and payment-related records, so it is sensitive and regulated, and the exploits are already being attempted in the logs, so likelihood is not theoretical but demonstrated