
WEEK 1 – DAY 7

WEEKLY REVISION + INTERVIEW THINKING

(Security-Focused, Easy-to-Explain, Interview-Ready)

PART 1: HIGH-LEVEL REVISION (Days 1–6)

Day 1 – Security Fundamentals

You learned:

- CIA Triad: Confidentiality, Integrity, Availability
- Threat vs Vulnerability vs Risk
- Attack surface
- Defense in depth
- Attacker vs Defender mindset

Key Interview Thought:
Security is about **reducing risk**, not achieving 100% safety.

Day 2 – Networking for Security

You learned:

- TCP/IP layers
- DNS working & attacks
- HTTP vs HTTPS
- Ports & protocols
- Firewalls
- TLS handshake
- Wireshark basics

Key Interview Thought:
Every attack travels through the **network**.

Day 3 – Web Architecture

You learned:

- Client–server model
- REST APIs

- HTTP methods
- Status codes
- Cookies vs headers
- Browser DevTools inspection

Key Interview Thought:
Client is **always untrusted**.

Day 4 – Authentication Basics

You learned:

- Password storage rules
- Hashing vs encryption
- bcrypt / Argon2
- Salting
- Login verification flow

Key Interview Thought:
Passwords are **verified, never decrypted**.

Day 5 – Authorization Models

You learned:

- Authentication vs Authorization
- RBAC, ABAC, ACL
- Least privilege
- Role-based access in Flask

Key Interview Thought:
Most breaches are **authorization failures**, not login failures.

Day 6 – Session vs Token Auth

You learned:

- Cookies & security flags
- Sessions
- JWT structure
- Access vs refresh tokens

- JWT login API

Key Interview Thought:
JWT improves scalability but increases security responsibility.

PART 2: EXPLAINING CONCEPTS ALOUD (VERY IMPORTANT)

You must practice **speaking**, not just reading.

Below are **model answers** you should practice saying.

---

1. Explain CIA Triad (Out Loud)

"The CIA triad defines the three core security goals:
Confidentiality ensures data is accessible only to authorized users,
Integrity ensures data is not modified without authorization,
and Availability ensures systems are accessible when needed.
Any security control maps to one or more of these."

---

2. Explain Session vs JWT (Out Loud)

"Sessions are server-side stored authentication states identified by a session ID, while JWTs are stateless tokens that carry user claims and are verified using a signature. Sessions are easier to invalidate, while JWTs scale better but require careful handling."

---

3. Explain RBAC Simply (Out Loud)

"RBAC grants permissions based on predefined roles, which simplifies authorization management but can become rigid in complex environments."

PART 3: INTERVIEW SHORT-ANSWER PRACTICE (CRITICAL)

These are **exact interview-style questions**.

Q1: "How does authentication fail?"

Short, Strong Interview Answer:

"Authentication fails when identity verification is weak or improperly implemented.
Common failures include storing passwords in plain text, using weak hashing algorithms, lack of rate limiting, insecure password policies, credential reuse, and improper session or token handling.
These failures allow attackers to bypass login controls or take over accounts."

Expanded Answer (If Interviewer Asks More):

“Authentication can fail due to technical issues like weak password storage or logical issues like missing brute-force protection. Token-based systems may also fail if tokens are leaked, stored insecurely, or lack expiration.”

Q2: “Why is JWT risky?”

Short, Strong Interview Answer:

“JWT is risky because it is self-contained and stateless. If a JWT is stolen, the attacker can use it until it expires. Additionally, JWT payloads are readable, logout is difficult to enforce, and improper storage or weak secrets can lead to token abuse.”

Expanded Answer (Interview Gold):

“JWT increases scalability but shifts security responsibility to token management. Risks include token leakage, long expiration times, storing tokens in localStorage, lack of rotation, and weak signing keys. Without careful design, JWT can increase attack impact.”

PART 4: COMMON FOLLOW-UP QUESTIONS (Be Ready)

If interviewer asks:

“When should we NOT use JWT?”

Answer:

“For traditional server-rendered applications where session invalidation and simplicity are more important than scalability.”

If interviewer asks:

“How do you secure JWT?”

Answer:

“By using short-lived access tokens, secure storage, HTTPS, strong secrets, token rotation, and refresh tokens.”