Unit 10 – API Security Requirements

An Application Programming Interface (API) acts as a digital intermediary that allows two different software applications to communicate with one another. It defines a set of protocols and tools that specify how software components should interact, enabling a "request and response" cycle where a client (such as a Python script) can retrieve or send data to a server without needing to understand the server's internal code.

The Ambulance Data Submission - FHIR API is an API used by the National Health Service (NHS) in the United Kingdom. It is designed to facilitate the secure transfer of emergency care records from ambulance service providers to central healthcare repositories. It leverages the Fast Healthcare Interoperability Resources (FHIR) standard to ensure that complex clinical data remains consistent and computable across different electronic health record systems.

Modern API security relies on the "Zero Trust" model, where no entity is trusted by default regardless of its location on the network. Core concepts include authentication (verifying identity), authorisation (enforcing specific access rights), and availability (protecting against Denial-of-Service attacks). In healthcare, these concepts are critical as APIs expand the attack surface, introducing risks such as unauthorised data exfiltration and "Broken Object Level Authorisation," where attackers manipulate IDs to access records they do not own (Mendes and Vilela, 2023).

Security Requirements Specification

- Identity and Access Management (IAM): Access must be restricted using OAuth 2.0 with OpenID Connect (OIDC). Scope-based permissions are essential to ensure the Python client only accesses the minimum necessary FHIR resources (Principle of Least Privilege).
- Data Encryption: All data in transit must be protected via TLS 1.3. For data at rest within SQL databases or JSON/XML files, AES-256 encryption is required to prevent data breaches following unauthorised physical or cloud access.
- Input Validation and Sanitisation: To prevent SQL Injection and XML External Entity (XXE) attacks, the API must enforce strict schema validation. Python scripts should use parameterised queries when interfacing with SQL and defused XML libraries to mitigate parsing vulnerabilities (Williams and Mclaughlin, 2022).
- Rate Limiting and Throttling: To prevent automated scraping and Denial of Service (DoS) attacks, the API must implement threshold-based rate limiting, ensuring that high-frequency requests from automated scripts are flagged and blocked.
- Audit Logging: Comprehensive logs of all FHIR transactions must be maintained to support forensic analysis and ensure compliance with healthcare data regulations (Hansen et al., 2021).

References

Hansen, S., Jensen, M. and Larsen, T. (2021) ‘Security challenges in FHIR-based health data exchange’, *Journal of Medical Systems*, 45(12), pp. 104–112.

Mendes, A. and Vilela, J. P. (2023) ‘Privacy-preserving data sharing in healthcare using FHIR’, *IEEE Access*, 11, pp. 4501–4515.

Williams, J. and Mclaughlin, S. (2022) ‘Mitigating injection risks in Python-based healthcare applications’, *International Journal of Information Security*, 21(3), pp. 289–305.