? EASY (Q1–Q10)

Q1. SAST stands for:
A. Secure Application Software Testing
B. Static Application Security Testing
C. System Application Security Tool
D. Software Analysis Security Technique

Q2. DAST analyzes applications during:
A. Design phase
B. Development phase
C. Runtime execution
D. Compilation

Q3. Which SDLC phase is SAST primarily associated with?
A. Deployment
B. Design
C. Development
D. Maintenance

Q4. Which testing approach does NOT require source code?
A. SAST
B. Code review
C. DAST
D. Static analysis

Q5. Browser-based security tools mainly analyze:
A. Server hardware
B. Client-side behavior
C. Database schema
D. Network cables

Q6. Which security objective is improved by early vulnerability detection?
A. Availability
B. Confidentiality
C. Integrity
D. All of the above

Q7. SAST tools primarily detect:
A. Runtime configuration issues
B. Network attacks
C. Code-level vulnerabilities
D. Physical threats

Q8. DAST tools interact with applications as:
A. Developers
B. Attackers
C. System administrators
D. Database users

Q9. Browser add-ons like security guards help in detecting:
A. Malware behavior
B. Suspicious web scripts
C. Database injections
D. OS vulnerabilities

Q10. False positives refer to:
A. Missed vulnerabilities
B. Correct vulnerability detection
C. Incorrect vulnerability alerts
D. Encrypted findings

□ MEDIUM (Q11–Q25)

Q11. SAST tools analyze which artifact?
A. Network traffic
B. Source code or binaries
C. User sessions
D. Server logs

Q12. DAST tools are best suited for detecting:
A. Business logic flaws
B. Runtime vulnerabilities
C. Source code syntax errors
D. Design flaws

Q13. White-box testing is most closely related to:
A. DAST
B. Black-box testing
C. SAST
D. Penetration testing

Q14. Black-box testing simulates:
A. Insider threats
B. Real-world attackers
C. Developers
D. Administrators

Q15. Which vulnerability is commonly detected by DAST but not SAST?
A. SQL syntax errors
B. Runtime authentication issues
C. Hardcoded credentials
D. Dead code

Q16. Browser-based security analysis mainly protects against:
A. Server-side injections
B. Client-side attacks
C. Physical threats
D. OS kernel exploits

Q17. SAST tools may fail to detect vulnerabilities caused by:
A. Poor logic flow
B. Runtime configuration
C. Insecure coding
D. Hardcoded secrets

Q18. DAST scanning limitations include:
A. No runtime visibility
B. False negatives
C. Inability to detect code flaws
D. All of the above

Q19. Which SDLC phase benefits MOST from SAST?
A. Deployment
B. Maintenance
C. Development
D. Incident response

Q20. Security testing integrated early in SDLC is known as:
A. Reactive security
B. Shift-left security
C. Defensive coding
D. Incident handling

Q21. Browser-based security add-ons typically monitor:
A. Network packets only
B. JavaScript execution and behavior
C. Server logs
D. Database queries

Q22. Which testing approach detects vulnerabilities without executing code?
A. DAST
B. Dynamic testing
C. SAST
D. Fuzzing

Q23. Reports generated by SAST tools usually include:
A. Network topology
B. Code location of flaws
C. Packet captures
D. User behavior logs

Q24. DAST tools primarily interact using:
A. Source files
B. HTTP/HTTPS requests
C. Binary patching
D. Kernel hooks

Q25. Which is a major challenge in automated security testing?
A. Performance
B. False positives
C. Encryption
D. Network latency

🟥 HARD (Q26–Q40)

Q26. SAST tools are limited in detecting vulnerabilities caused by:
A. Insecure APIs
B. Runtime configuration errors
C. Weak cryptography
D. Hardcoded credentials

Q27. DAST tools may miss vulnerabilities due to:
A. Limited attack surface coverage
B. Lack of source code
C. Encryption usage
D. Antivirus interference

Q28. Which testing approach provides maximum visibility but requires trust?
A. Black-box
B. Grey-box
C. White-box
D. Dynamic testing

Q29. Combining SAST and DAST provides:
A. Redundant testing
B. Complete security coverage
C. Increased false positives
D. Reduced cost

Q30. Browser-based detection tools rely heavily on:
A. Signature matching
B. Heuristic and behavioral analysis
C. Network firewalls
D. Source code access

Q31. Insecure framework configurations are best detected by:
A. SAST only
B. DAST only
C. Both SAST and DAST
D. Manual testing only

Q32. Automated security tools struggle MOST with detecting:
A. SQL Injection
B. XSS
C. Business logic flaws
D. Misconfigurations

Q33. False negatives are dangerous because they:
A. Overestimate risk
B. Underestimate security posture
C. Crash applications
D. Increase scan time

Q34. SAST tools often generate more false positives because they:
A. Execute applications
B. Analyze code paths conservatively
C. Use runtime data
D. Ignore encryption

Q35. Browser-based security add-ons enhance security by:
A. Blocking server attacks
B. Preventing user interaction
C. Alerting users to malicious scripts
D. Encrypting traffic

Q36. Secure SDLC emphasizes:
A. Security only at deployment
B. Continuous security testing
C. Network security only
D. Post-incident analysis

Q37. Which testing approach is BEST for CI/CD pipelines?
A. Manual testing
B. SAST
C. Social engineering
D. Physical testing

Q38. DAST tools operate at which testing level?
A. Code level
B. Binary level
C. Application behavior level
D. OS kernel level

Q39. Which vulnerability is most likely detected via browser-based analysis?
A. Server-side SQL Injection
B. Client-side XSS
C. Buffer overflow
D. Kernel exploit

Q40. The primary benefit of combining browser-based tools with DAST is:
A. Reduced scanning time
B. Improved client-side attack detection
C. Source code access
D. Hardware security