Technology Update

Building Secure Applications

Cybersecurity Best Practices for Modern Architecture

Edited by **C.J. ANTONY**

In today's hyper-connected world, cybersecurity is essential in application architecture. As businesses grow reliant on digital platforms, securing applications from cyber threats is critical for protecting data, maintaining trust, and ensuring business continuity. Cybersecurity in application architecture involves safeguarding hardware, software, and data from unauthorized access, breaches, and attacks. Without proper security, hackers can exploit vulnerabilities, steal data, or disrupt services. Cyberattacks can result in financial loss, legal repercussions, and reputational damage. Securing application architecture helps prevent these risks, ensuring systems remain resilient against emerging threats.

Secure System Development Life-Cycle (SSDLC)

To effectively secure applications, organizations must adopt a proactive approach, starting from the very beginning of development. Integrating security into every phase of the Software Development Life Cycle (SDLC) is the primary step to ensure security of applications. Instead of treating security as an afterthought, it should be embedded in design, development, testing, and deployment.

- **Early Security Integration:** Address security concerns from the design stage to identify risks and vulnerabilities before development begins.
- **Static / Dynamic Application Security Testing:** Use SAST and DAST tools to detect and fix vulnerabilities in code and applications.
- **Threat Modelling:** Conduct early threat modelling to anticipate security risks.
- **Security Training for Developers:** Train developers and stakeholders in secure coding and keep them updated on vulnerabilities like the OWASP Top Ten.

B. Kalaimani
Scientist-D
kmani@nic.in

Applications expose resources to targeted users and potential attackers. A secured architecture is the foundation of any application's defence against cyber threats. It provides a proactive and strategic defence, anticipating and mitigating potential risks before they manifest. It ensures a robust framework for protecting sensitive data and maintaining the integrity of the application as a whole.

Role-Based Access Control (RBAC)

Role-based access control is a method for managing user access to systems and resources based on a user's role or job function. RBAC allows IT administrators to assign roles to users with the appropriate permissions so that they are allowed access only to the information they need to know or perform their job duties. Access control is essential to protect sensitive parts of an application.

- **Limit Access by Role:** Implement RBAC to ensure users only access necessary data.
- **Backend and Frontend Enforcement:** Apply access controls at both user interface and backend levels.
- **API Restrictions:** Block unauthorized API access using reliable gateways.

Secure Authentication

Authentication confirms that only the right people with the right permissions can get access to the applications and data. Proper authentication mechanisms ensure sensitive data remains protected. Some of the tools and technologies that can be employed for this purpose are:

- **Multi-Factor Authentication (MFA):** Strengthen authentication with multiple forms of identity verification like DSC, OTP (Time-Based One-Time Password), FIDO2, Tokens, or app-based authenticators.
- **Password Policies:** Enforce long and strong passwords, with minimum 12 characters including special characters, numbers, and case variations.
- **Secure Session Management:** Implement secure session handling, automatic timeouts, and safe password reset mechanisms.
- **Password Communication:** Any change in password should be intimated to the user by SMS.

Proper Security Configuration

Security configuration is the process of setting up security controls and parameters for computer systems, networks, or software applications to reduce security risks. Proper security configuration is the key to security of any enterprise application. Incomplete and incorrect configurations can leave applications vulnerable, resulting in unauthorized access and exploitation.

- **Remove Default Settings:** Replace default credentials and configurations with secure alternatives.
- **Custom Error Pages:** Use custom error pages to obscure technical details.
- **Limited Privilege:** Provide only the required access to the users over the folders, databases and other resources.

Regular Patching and Vulnerability Management

Use of outdated software components in applications makes the application susceptible to cyber-attacks as the miscreants exploit the known vulnerabilities in these components. Regular and continuous patching to keep the components up-to-date reduces the risk of an application falling victim to attacks.

- **Patch Updates:** Apply security patches and updates regularly to frameworks and software.
- **Vulnerability Scanning:** Continuously scan applications for vulnerabilities.

- **Monitor Alerts:** Stay informed about vulnerabilities and patch them promptly.

Secure Communication Protocols

Secure communication protocols are rules and procedures that ensure that data transmission across a network is secure. A secure connection is one that uses encryption protocols to protect the data being transferred. Secured connections protect data from man-in-the-middle attacks, and ensures the data has not been tampered with during transit.

- **Enforce HTTPS:** Encrypt data between clients and servers using strong SSL/TLS configurations.
- **Disable Insecure Protocols:** Turn off outdated protocols like SSLv3.

Logging, Monitoring, and Incident Response

Logging, monitoring, and incident response are the three important activities for detecting and responding to security incidents. Log monitoring is essential for incident response as it enables organizations to detect and analyze security incidents, system failures, and operational issues. Hence, these activities should be envisaged while architecting the application itself.

- **Comprehensive Logging:** Record security-related events such as access attempts and system errors.
- **Log Security:** Ensure logs are securely stored and access is restricted to authorized personnel.
- **Incident Response Plan:** Develop formal procedures for handling security incidents.

DevSecOps Integration

DevSecOps is a framework that integrates security into every stage of the software development lifecycle. It stands for development, security, and operations. Each term defines different roles and responsibilities of software teams when they are building software applications. Integrating security into the DevOps process ensures continuous protection.

- **Security as Code:** Use tools like SonarQube and OWASP Dependency-Check in the CI/CD pipeline.
- **Container Security:** Secure containerized applications by using minimal base images and performing regular scans.

Database Security

Database security is crucial for protecting sensitive data from accidental and intentional threats. Data being the new oil, modern day hackers focus on data exfiltration or encryption through some ransomware. Compliance to legal frameworks like data protection laws also necessitates secure database management.

- **Use Stored Procedures:** Implement stored procedures and parameterized queries for database interactions instead of direct SQL queries.
- **Limited access to Data:** Access Privilege may be need to know basis, like Read Only access for Reports and Dashboard.
- **Secure file names and folders:** Do not keep files in directly accessible directories to prevent unauthorized access through the web server. Generate unique filenames using Global Unique Identifier (GUID) to prevent guessing and overwriting files.
- **Encryption at Rest:** Encrypt data using strong Keys to protect sensitive information stored in the database. Consider field level encryption for highly sensitive data like credit card information. Implement a secure key management system to handle encryption keys.

Traditional Security Approach	Modern Security Approach
Firewalls & VPNs	Zero Trust & Micro-Segmentatio
Basic Authentication	Multi-Factor & Passwordless Authentication
Patch Management	Continuous Security Monitoring & AI-driven Threat Intelligence
Perimeter-Based Security	Identity-Centric & Context-Aware Security
Manual Security Checks	Automated Security Testing & DevSecOps
Centralized Data Access	Role-Based & Least Privilege Access Control

API Security

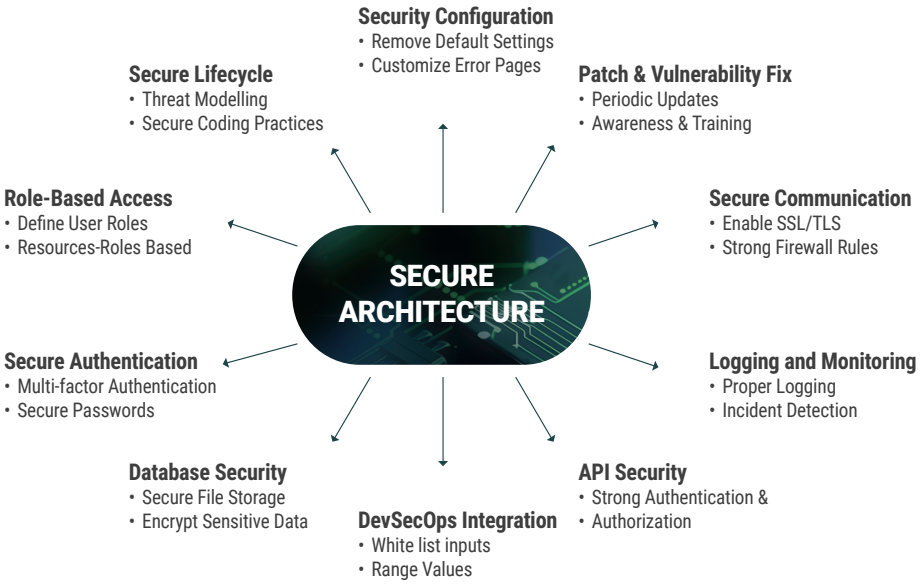
APIs (Application Programming Interface) are the communication channels between software systems. Protecting APIs from attacks deserves attention while architecting the application. API security is important because it protects sensitive data and prevents unauthorized access to APIs.

- **Strong Authentication and Authorization:** Implement robust authentication mechanisms to ensure that only authorized users and applications can access API. Enforce role-based access control (RBAC) to limit user permissions based on their roles.
- **Encryption in Transit:** Use appropriate protocols, like TLS (Transport Layer Security), to encrypt data being transmitted between the client and the server, as well as between different database components.

Conclusion

Implementing cybersecurity best practices is crucial for protecting systems, data, and users from evolving threats. By securing all stages of development, from design to deployment, organizations can prevent vulnerabilities, reduce the risk of attacks, and maintain compliance. Organisations may also formulate a Cyber Security Policy that govern how the information systems, data, and resources are protected from internal and external threats. The policy may be reviewed periodically for maintaining an effective cybersecurity strategy to ensure that the policies remain relevant, up-to-date, and aligned with evolving threats, technologies, and compliance regulations. These practices foster trust, reliability, and resilience in today's digital world.

Fig 10.1 **Benefits of Secure Architecture**



Contact for more details

State Informatics Officer
NIC, Tamil Nadu State Centre
E2-A, Rajaji Bhavan, Besant Nagar, Chennai-600090
Email: sio.tn@nic.in, Phone: 044-24466495