SAMM Assessment spreadsheet

Software Development Plan for NexFlow by NexSoft

1.Governance

Strategy & Metrics:

- **Create & Promote**: Develop a strategic roadmap focusing on secure embedded systems for industrial applications. Establish long-term security goals based on product-specific requirements.
- **Measure & Improve**: Implement a feedback loop to monitor metrics like incident detection, downtime reduction, and system performance. Analyze these metrics periodically to improve NexFlow's security posture.

Policy & Compliance:

- **Policy & Standards**: Establish security policies to guide NexFlow’s software development. Define encryption standards, data access controls, and security protocols.
- **Compliance Management**: Develop a compliance monitoring framework to ensure that NexFlow meets industry and regulatory standards at all times.

Education & Guidance:

- **Training & Awareness**: Provide regular cybersecurity training for all employees. Focus on emerging threats like phishing attacks and password management to maintain a high level of awareness across the company.
- **Organization & Culture**: Foster a culture where cybersecurity is a shared responsibility. Ensure that senior management emphasizes security awareness through regular communications and leadership initiatives.

2.Design

Threat Assessment:

- **Application Risk Profile**: Conduct a comprehensive risk assessment, analyzing how NexFlow integrates with **industrial networks**. Identify potential vulnerabilities in data transmission between **Edge Gateways** and the **Cloud Platform**.
- **Threat Modeling**: Develop threat models based on NexFlow’s specific environment. Simulate potential cyberattacks on connected devices and data flow pathways.

Security Requirements:

- **Software Requirements**: Specify that all data between Edge Gateways, Cloud Platform, and industrial networks must be encrypted and authenticated. Implement multi-factor authentication for accessing sensitive system components.
- **Supplier Security**: Ensure that any third-party software or libraries integrated into NexFlow undergo rigorous security testing and are compliant with NexSoft's security policies.

Secure Architecture:

- **Architecture Design**: Adopt a modular design for NexFlow that isolates critical system components. Use a defense-in-depth approach to protect data flow between industrial networks and the cloud.
- **Technology Management**: Ensure secure connectivity across various industrial networks, using end-to-end encryption for data transfers. Utilize secure protocols like TLS and HTTPS for cloud connectivity.

3. Implementation

Secure Build:

- **Build Process**: Integrate automated security testing into the CI/CD pipeline to catch potential vulnerabilities early. Run static analysis on all code before deployment.
- **Software Dependencies**: Ensure that all third-party libraries and dependencies are vetted and regularly updated to mitigate known vulnerabilities.

Secure Deployment:

- **Deployment Process**: Implement secure deployment processes that involve containerization and microservices to isolate system functions. Use role-based access control (RBAC) for sensitive deployment operations.
- **Secret Management**: Use a secure vault to manage sensitive credentials and secrets such as API keys and authentication tokens.

Defect Management:

- **Defect Tracking**: Track security-related defects through a centralized issue tracker, ensuring prompt remediation of vulnerabilities found during the build phase.
- **Metrics & Feedback**: Continuously evaluate defect metrics to assess common failure points and areas of improvement.

4.Verification

Architecture Assessment:

- **Architecture Validation**: Conduct periodic security assessments to ensure that the modular design of NexFlow is resilient against cyberattacks. Validate that encryption and access control mechanisms function as intended.
- **Architecture Compliance**: Ensure the architecture complies with industry standards and perform regular audits.

Requirements-driven Testing:

- **Control Verification**: Implement **control verification** mechanisms to check that all security controls, such as **MFA** and **encryption**, work as expected across all components.
- **Misuse/Abuse Testing**: Perform **penetration testing** and simulate cyberattacks to evaluate NexFlow's ability to withstand misuse or abuse scenarios .

Security Testing:

- **Scalable Baseline**: Perform continuous security testing, using Dynamic Application Security Testing and Static Application Security Testing tools for ongoing assessment.
- **Deep Understanding**: Conduct deep analysis of critical components like edge gateway communication and data encryption to ensure that no vulnerabilities are present in the system's core.

5.Operations

Incident Management:

- **Incident Detection**: Utilize Security Information and Event Management tools to monitor real-time activity and detect anomalies across NexFlow's system components.
- **Incident Response**: Develop an incident response plan that includes identifying the breach, mitigating the damage, and restoring normal operations with minimal disruption.

Environment Management:

- **Configuration Hardening**: Harden configurations for all critical systems, including cloud and industrial networks, to minimize attack vectors. Regularly review firewall and intrusion detection system settings.
- **Patch & Update**: Implement automated patching systems for third-party libraries and software to minimize risks from unpatched vulnerabilities.

Operational Management:

- **Data Protection**: Ensure that sensitive data are encrypted at rest and in transit. Implement a secure data retention policy, ensuring compliance with GDPR and CCPA.
- **Legacy Management**: Develop a process to manage any legacy systems integrated into NexFlow, ensuring that outdated software or hardware does not compromise security.