

# CRA & NIS2: Urgent Compliance for Embedded Systems

- The **Cyber Resilience Act (CRA)** and **NIS2 Directive** establish comprehensive cybersecurity requirements for embedded systems with digital elements
- Manufacturers have **36 months** from December 2024 to comply with CRA; NIS2 national laws effective from October 2024
- Embedded ARM64 systems with WiFi/Bluetooth connectivity require **security-by-design** approach to meet compliance
- Non-compliance risks include significant fines, market restrictions, and reputational damage



## KEY INSIGHT

Securing U-boot, Linux kernel, and APT package repositories is essential for CRA/NIS2 compliance and must be addressed by 2026

# New Regulations Mandate Robust Cybersecurity

## CRA Requirements

- Comprehensive security risk assessments
- Secure-by-design development
- Vulnerability monitoring & patching
- Technical documentation

## NIS2 Requirements

- Enhanced cybersecurity capabilities
- Risk management measures
- Supply chain security policies
- Incident reporting obligations

## KEY INSIGHT

Both regulations hold top management accountable for cybersecurity compliance, elevating security to a board-level concern for embedded system manufacturers

## Regulatory developments in cybersecurity



# ARM64 Systems Face Unique Security Challenges

## Architecture Considerations

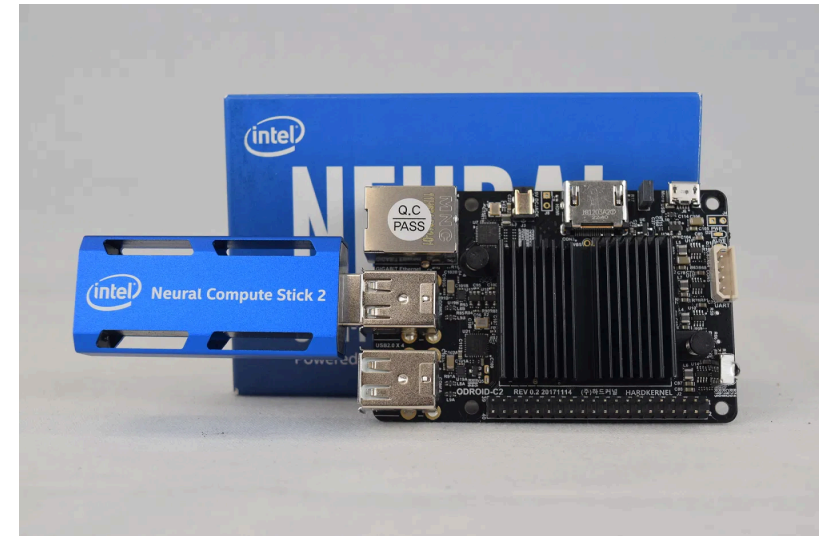
- 64-bit architecture with TrustZone for hardware-based isolation
- Memory protection and Execute Never (XN) capabilities must be properly configured

## Security Constraints

- Limited resources for comprehensive security implementations
- Long lifecycle requiring extended support and vulnerability management
- WiFi and Bluetooth interfaces significantly expand attack surface

### KEY INSIGHT

Embedded ARM64 systems require a layered security approach starting from boot process through kernel to package management



# Secure Boot Establishes Trust from Hardware

## 1 Hardware Root of Trust

Immutable boot ROM code verifies first-stage bootloader using cryptographic signatures

## 2 Chain of Trust

Each stage verifies the next component before execution, creating an unbroken chain of trusted software

## 3 Cryptographic Verification

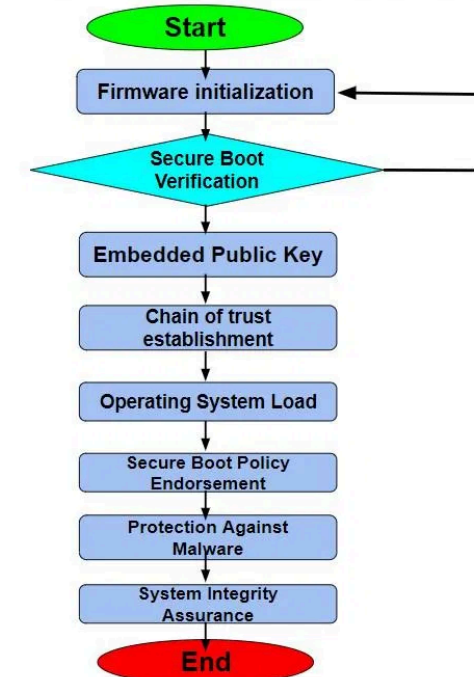
RSA-2048 or stronger signatures with secure key storage in hardware (TPM/HSM)

## 4 Protection Mechanisms

Immutable boot firmware, rollback prevention, and debug port protection

### KEY INSIGHT

Secure boot is the foundation of CRA/NIS2 compliance, preventing unauthorized code execution and establishing system integrity from power-on



# U-Boot Security Hardening


## **Secure Boot Implementation:**

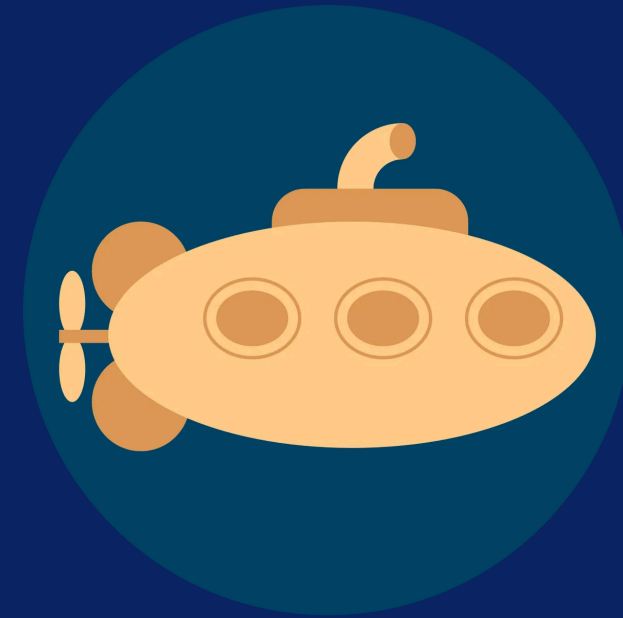
- Signed FIT (Flattened Image Tree) images
- Hardware cryptographic verification
- Chain of trust establishment

## **U-Boot Hardening Techniques:**

- Command whitelisting to reduce attack surface
- Self-overwriting protection
- CLI access prevention in production
- Kernel command-line protection

 **Regular Updates:** Vulnerability monitoring and patching

 **CRA Requirement:** Consider disabling JTAG interfaces in production



# U-Boot

# Linux Kernel Hardening

## 🛡️ Memory Protection:

- Kernel Address Space Layout Randomization (KASLR)
- Execute Never (XN) bit implementation
- Memory Tagging Extension (MTE) for ARM64

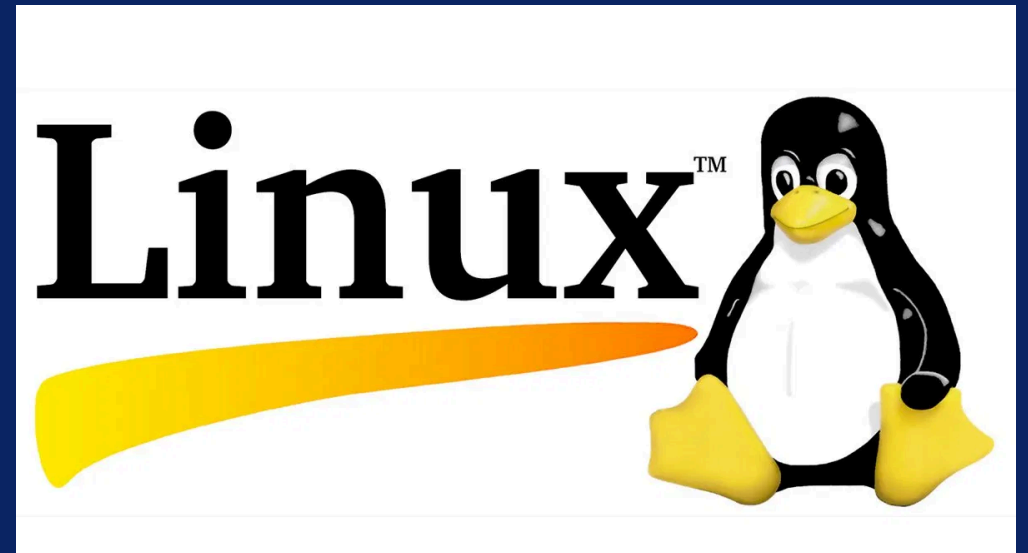
## 🔒 Access Control:

- SELinux/AppArmor Mandatory Access Control
- Seccomp for syscall filtering

## ✅ Integrity Protection:

- dm-verity for root filesystem verification
- Signed kernel modules enforcement

## 🔑 Configuration: Minimize attack surface by disabling unnecessary features



# APT Package Repository Security

## Supply Chain Security Risks:

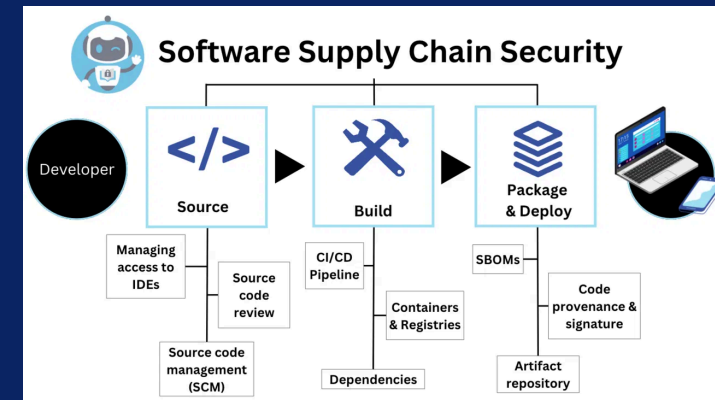
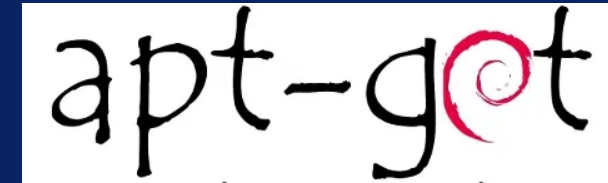
- Package tampering and malicious code injection
- Man-in-the-middle attacks during package downloads
- Dependency confusion attacks

## Key Security Measures:

- Cryptographic signing of packages and repositories
- HTTPS transport for all repository access
- Dedicated private repositories for embedded systems

## Regulatory Requirements:

- CRA: Software supply chain security measures
- NIS2: Risk management for critical infrastructure



# Certificate and Key Management

## 🔑 GPG Key Management:

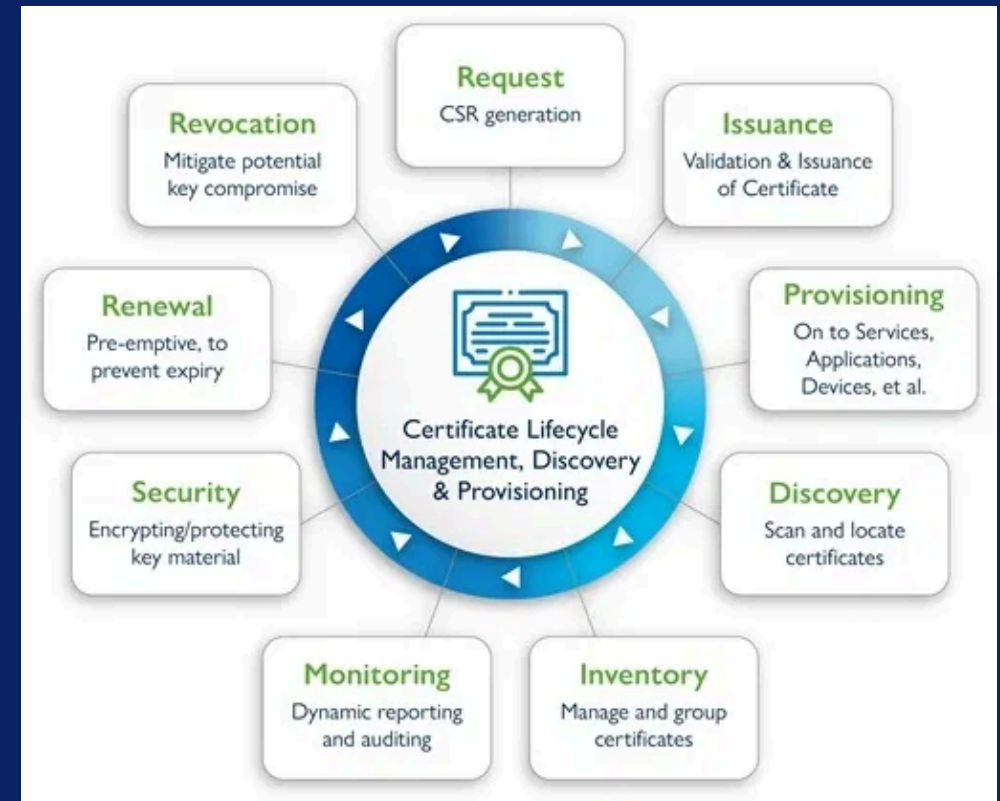
- Use RSA 4096-bit or higher for signing keys
- Store private keys in HSMs or smart cards
- Implement regular key rotation (annually)

## 🛡️ Modern APT Security:

- Avoid deprecated apt-key; use  
/etc/apt/trusted.gpg.d/
- Implement keyring isolation for different repositories
- Use signed-by option in source entries

🔒 **Offline Signing:** Sign packages on air-gapped systems

⚠️ **Revocation:** Establish clear procedures for compromised keys





# Secure Repository Server Configuration

## **Server Hardening:**


- Minimal installation with only necessary services
- Regular security updates and patches
- Network segmentation and firewall rules

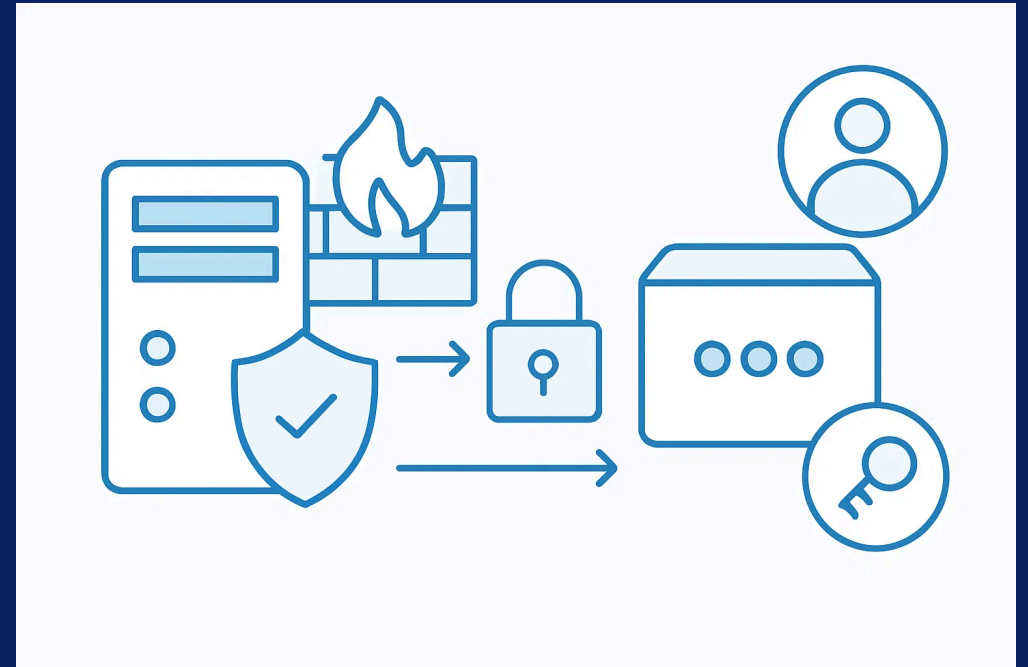
## **Transport Security:**

- HTTPS with TLS 1.3 for all repository access
- Strong cipher suites and certificate validation
- Certificate Authority (CA) management

## **Access Control:**

- Multi-factor authentication for administrators
- Role-based access control (RBAC)
- IP-based restrictions for critical operations

 **Audit Logging:** Comprehensive logging of all repository operations



# Client-Side APT Security

## ✓ Package Verification:

- Always verify package signatures
- Reject unsigned or invalidly signed packages
- Configure `APT::Get::AllowUnauthenticated` "false"

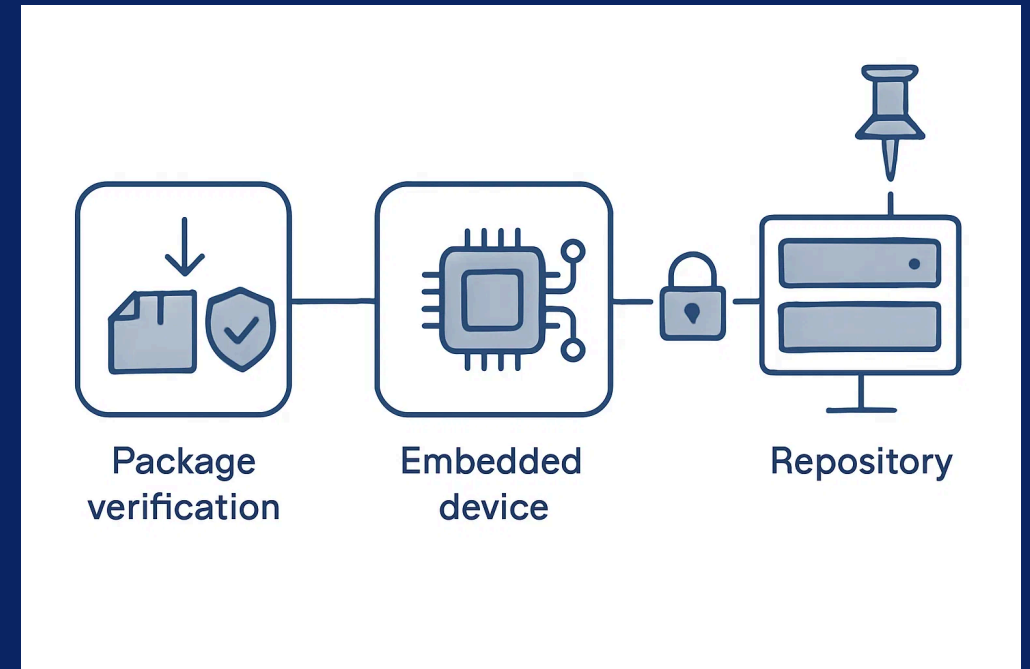
## 🔒 Transport Security:

- Enforce HTTPS for all repository connections
- Validate TLS certificates properly

## 📌 APT Pinning:

- Implement pinning to prioritize trusted repositories
- Prevent package downgrades with  
APT::Get::AllowDowngrade "false"

🔄 **Regular Updates:** Ensure client systems update trusted keyrings



# Compliance Validation and Reporting

## 📋 Assessment Methodologies:

- Security gap analysis against CRA/NIS2 requirements
- Penetration testing of embedded systems
- Vulnerability scanning of firmware and packages

## 📄 Documentation Requirements:

- Software Bill of Materials (SBOM)
- Risk assessment reports
- Security implementation evidence

## 📈 Continuous Monitoring:

- Automated security testing in CI/CD pipeline
- Regular compliance audits

⚠️ **Incident Response:** Required procedures for security incidents



# Conclusion and Next Steps

## ✓ Key Takeaways:

- CRA and NIS2 compliance is mandatory by 2026
- Secure boot and kernel hardening are foundational
- APT repository security is critical for supply chain integrity

## ☰ Implementation Roadmap:

- Conduct security assessment of current systems
- Develop secure boot implementation plan
- Establish secure APT infrastructure with proper key management
- Implement continuous monitoring and compliance validation

📅 **Timeline:** Begin implementation now to ensure compliance by 2026

