NPCC
National Police Chiefs' Council

POLICE DIGITAL SERVICE

CYBER GUIDELINE DOCUMENT

Internet Of Things (IoT)

ABSTRACT:
This guideline serves as a foundational resource for embedding Internet Of Things devices through the application of secure by design principles, helping members of the community of trust to mitigate risks to integrity, confidentiality, and availability.

ISSUED	September 2025
PLANNED REVIEW DATE	September 2026
DISTRIBUTION	Community Security Policy Framework Members

POLICY VALIDITY STATEMENT
This guideline is due for review on the date shown above. After this date, this document may become invalid.

Cyber guideline users should ensure that they are consulting the currently valid version of the documentation.

CONTENTS

Community Security Policy Commitment

National Policing and its community members recognise that threats to policing information assets present significant risk to policing operations. National Policing and its community members are committed to managing information security and risk and maintaining an appropriate response to current and emerging threats, as an enabling mechanism for policing to achieve its operational objectives whilst preserving life, property, and civil liberties.

This guideline in conjunction with the National Policing Community Security Policy Framework and associated documents sets out National Policing requirements described in the Secure development, network Security, physical asset management and application security policy areas.

Introduction

In an era where connectivity is revolutionising and enabling policing, the deployment of Internet of Things (IoT) devices and connected vehicles presents both groundbreaking opportunities and critical security challenges. As these technologies integrate into sensitive and operational environments, ensuring their resilience against cyber threats is paramount.

This guideline serves as a foundational resource for embedding secure by design principles, helping members of the community of trust to mitigate risks to integrity, confidentiality, and availability.

By implementing robust cyber security controls, members can safeguard their digital, data and technology systems, protect operations, and ensure trust in an increasingly interconnected world.

Owner

National Chief Information Security Officer (NCISO).

Purpose

The purpose of this guideline is to detail the recommended approach to securely provisioning, assess, deploy and decommission devices, vehicles, appliances, and other objects that are embedded with sensors, software, and network connectivity, enabling them to exchange data with other devices and systems over the internet.

The guideline follows the Secure Development (Secure by Design) approach to ensure that risks are identified and managed throughout the lifecycle of IOT which are used in policing environments.

Audience

Policing Community of Trust – specifically, but not limited to, Senior Information Risk Owners (SIRO), Information Security Officers (ISO), Project Management Office (PMO) and Information Asset Owners (IAO).

This guideline is aimed at:

- Information Asset Owners (IAOs), Platform Asset Owners (PAOs), and Senior Information Risk Owners (SIROs).
- Project Management Office (PMO) and business change leads.
- Architects, system designers and engineers.
- Information & Cyber risk practitioners and managers.
- Staff across policing who are responsible for the selection, development or deployment of IT systems or applications either on behalf of national policing or at a local force level.
- Suppliers acting as service providers or developing products or services for policing.
- Auditors providing assurance services to policing.

Scope

1. The principles and methodology of this guideline are the foundation for National policing IT systems, applications, or service implementations. The requirements will be applied to new and existing installations.
2. This standard is applicable to any infrastructure, system, application, or IT solution that processes or stores policing information assets.
3. The security control requirements laid out in this guideline are vendor agnostic and applicable for all IT systems, applications, or service implementations that are provisioned for policing community of trust use.

Introduction

The IoT (Internet of Things) Security Guidelines are designed to help organisations secure their IoT infrastructures by embedding security at every stage of the device lifecycle. From design to deployment, continuous monitoring, and decommissioning, the target is to ensure that IoT devices and data remain secure against potential threats.

Internet of things (IoT) devices fall into 3 categories

- Sensors, which gather data
- Actuators, which effect actions
- Gateways, which act as communication hubs, and may also implement some automation logic

Devices identified as IoT

- CCTV, Access control systems, fire alarms, door locks
- Wearable tech
- Voice activated products
- Health monitors
- IoT SIM or eSIM

Connected devices

- Vehicles with connected technology
- Commercial and licensed drones **– see NPCC briefing note 22nd February 2023**
- Enforcement technology

The Secure By Design Approach to managing IoT

1. Stakeholder Engagement and Collaboration

Objective

Engaging with all relevant stakeholders early in the process is essential for ensuring IoT systems are secure from the outset.

Approach

- Identify Stakeholders: Include IT, security, operational staff, IoT product suppliers, locals forces and/or LEA and legal teams. Ensure that stakeholders understand their roles in implementing security controls.
- Collaborate Continuously: Regular collaboration is essential to align stakeholder goals with IoT security standards and project outcomes.
- Use a RACI Matrix: Define roles and responsibilities for securing IoT devices and infrastructures using a RACI matrix. This ensures accountability for key tasks like device management, vulnerability scanning, and data protection.

Example: When deploying IoT-based city systems, collaborate law enforcement agency, and IoT device manufacturers to ensure comprehensive security coverage.

2. Clarifying Roles and Responsibilities

Objective

Assign clear responsibilities for each task related to IoT security.

Approach

- Develop a RACI Matrix: For each IoT system component, define who is responsible, accountable, consulted, and informed. Ensure that all aspects of security—such as device authentication, patch management, and incident response—are covered.

Example: For an IoT system, assign roles such as device manufacturer (Responsible for firmware updates), local force IT team (Accountable for patch deployment), and the cybersecurity team (Consulted on security policies).

3. Project Scope and Deliverables

Objective

Establish clear boundaries for the IoT project and ensure that security requirements are included from the start.

Approach

- Define IoT System Scope: Identify the number of IoT devices, their locations, data flows, and connections to other systems. Include security features, such as encryption, device identity management, and network segmentation.
- Set Clear Deliverables: Include milestones like secure onboarding of IoT devices, regular updates, and periodic security audits.

Example: In a smart office project, define which devices will be connected to the network, how they will be secured, and how firmware updates will be managed.

4. IoT Security Requirements

Objective

Establish security measures that ensure the protection of IoT devices, data, and systems.

4.1. Feasibility, Appraise & Select, Define

- Pre-defined IoT Security Requirements: Ensure that IoT systems meet security standards such as OWASP IoT Top Ten, NIST IoT guidelines, or the European Union Agency for Cybersecurity (ENISA) recommendations.
- Threat Profiling: Conduct threat profiling specific to IoT environments, assessing risks such as device tampering, denial-of-service (DoS) attacks, or data breaches.
- Business Impact Assessment (BIA): Include a BIA to evaluate the potential consequences of security breaches in IoT systems, considering device tampering or service outages.

Example: A Business Impact Assessment for a smart grid might assess the potential disruption of critical utilities in the event of a DoS attack.

4.2. Deliver

4.2.1 Secure Design Principles

- Security by Design for IoT: Embed security from the outset by using principles like secure boot, firmware integrity checks, encryption of communications, and device authentication.
- Device Identity and Authentication: Ensure that each IoT device has a unique identity and uses strong authentication mechanisms to connect securely to the network.

4.2.2 Managed Change Control

- IoT Firmware and Software Updates: Implement a secure and managed change control process for firmware and software updates. Ensure that all updates are securely signed and verified before deployment.

4.2.3 External Software Components

- Third-party Components: Vet third-party software and hardware components for security vulnerabilities. Ensure compliance with relevant IoT security standards.

4.2.1. Asset Register Management

- Maintain an IoT Asset Register: Keep a detailed inventory of all IoT devices, including software versions, patches, and locations.

4.2.2. Resilient IoT System Design

- Resilient Design: Ensure IoT systems can withstand attacks by utilising techniques like redundancy, secure device communication, and continuous monitoring.

4.2.3. Coding Security

- Secure Coding for IoT: Follow secure coding guidelines, especially for embedded devices, ensuring firmware updates are regularly tested and vulnerabilities addressed.

4.2.4. System Functionality and Security

- Ensure IoT Functionality and Security: Continuously test IoT devices to ensure that they meet both functional and security requirements. Perform penetration testing to identify potential vulnerabilities.

4.2.5. Protecting Test Data

- Data Protection in Testing: Avoid using real data (especially Personally Identifiable Information) in IoT testing environments. Use synthetic or anonymized data for testing purposes.

System Testing and Live Environment Integration

- IoT Security Testing: Before deployment, ensure that IoT devices and networks are tested rigorously in a simulated live environment to identify potential security weaknesses.

5. Operate, Embed & Close

5.1. Change Management for IoT Systems

- Managed Change Control: Ensure that IoT system changes, such as new devices or updates, follow a controlled process to avoid introducing new vulnerabilities.

5.2. Managing IoT Assets Throughout Lifecycle

- Lifecycle Management: Maintain a secure asset management lifecycle, ensuring that IoT devices are tracked from procurement to decommissioning. Implement regular maintenance, updates, and security checks.

5.3. Secure Disposal of IoT Devices

- Disposal Protocols: Develop procedures to securely decommission and dispose of IoT devices, ensuring that sensitive data is erased before disposal.

Example: When decommissioning IoT cameras, ensure that any stored video data is securely erased, and the device is physically destroyed or disposed of in compliance with relevant regulations.

Risk Management controls for IoT devices

Aspect	Risk controls
Asset management NIST 800-53 V1.1, V2.0 ID.AM-01.ID.IAM-02	• Maintain inventories for all types of hardware, including IT, IoT, OT, and mobile devices • Constantly monitor networks to detect new hardware and automatically update inventories • Maintain inventories for all types of software and services, including commercial-off-the-shelf, open-source, custom applications, API services, and cloud-based applications and services • Constantly monitor all platforms, including containers and virtual machines, for software and service inventory changes • Maintain an inventory of the organization's systems
Secure development NIST 800-53 V1.1	Design, develop, deploy and configure applications and infrastructures such that Cloud Service Providers (CSP) and Cloud Services Consumers (CSC) (tenant) user access and intra-tenant access is appropriately segmented and segregated, monitored and restricted from other tenants. **Auditing Guidance** • Review evidence to verify that the design and development of applications and infrastructure ensure appropriate best practices such as hardening, segmentation, and segregation is incorporated and the shared responsibility model between the CSP and CSC is maintained. • Review evidence to verify that the deployment and configuration of applications and infrastructure follow appropriate hardening, segmentation, and segregation is incorporated and the shared responsibility model between the CSP and CSC is maintained. • Review evidence to determine that segmentation and segregation is monitored. • Review evidence to determine that the tenants are isolated from each other.

Aspect	Risk controls
Monitoring and vulnerability management NIST CSF 800-53 v2.0 REV 5	a. Monitor and scan for vulnerabilities in the system and hosted applications [Assignment: *organization-defined frequency and/or randomly in accordance with organization-defined process*] and when new vulnerabilities potentially affecting the system are identified and reported; b. Employ vulnerability monitoring tools and techniques that facilitate interoperability among tools and automate parts of the vulnerability management process by using standards for: • Enumerating platforms, software flaws, and improper configurations. • Formatting checklists and test procedures; and • Measuring vulnerability impact; • Analyse vulnerability scan reports and results from vulnerability monitoring; Remediate legitimate vulnerabilities [Assignment: *organization-defined response times*] in accordance with an organizational assessment of risk; c. Share information obtained from the vulnerability monitoring process and control assessments with [Assignment: *organization-defined personnel or roles*] to help eliminate similar vulnerabilities in other systems; and d. Employ vulnerability monitoring tools that include the capability to readily update the vulnerabilities to be scanned. Note: traditional vulnerability management tools designed for personal computer / servers may not effectively address IoT vulnerabilities. **See NCSP Vulnerability management standard**
Classification of Data	• Apply the UK Government security classification policy and document • Assess every item of data stored, processed, transmitted or received by a device and apply a data classification rating to it. Consider that collections of data may be more sensitive than individual items and so may be classified differently. • Ensure the security design protects against unauthorised viewing, changing or deletion, to at least its classification rating or higher. • When documenting the security design, also document the data items, their classification and the security design features that protect them.

Aspect	Risk controls
Physical security	• Any interface used for administration or test purposes during development should be removed from a production device, disabled or made physically inaccessible. • All test access points on production units must be disabled or locked, for example by blowing on-chip fuses to disable JTAG. • If a production device must have an administration port, ensure it has effective access controls, e.g. strong credential management, restricted ports, secure protocols etc. • Make the device circuitry physically inaccessible to tampering, e.g. epoxy chips to circuit board, resin encapsulation, hiding data and address lines under these components etc. • Provide secure protective casing and mounting options for deployment of devices in exposed locations. • To identify and deter access within the supply chain, consider making the device and packaging "tamper evident". • For high-security deployments, consider design measures such as active masking or shielding to protect against side-channel attacks.
Device secure boot	• Make sure the ROM-based secure boot function is always used. Use a multi-stage bootloader initiated by a minimal amount of read-only code (typically stored in onetime programmable memory). • Use a hardware-based tamper-resistant capability (e.g. a microcontroller security subsystem, Secure Access Module (SAM) or Trusted Platform Module (TPM) to store crucial data items and run the trusted authentication/cryptographic functions required for the boot process. Its limited secure storage capacity must hold the read-only first stage of the bootloader and all other data required to verify the authenticity of firmware. • Check each stage of boot code is valid and trusted immediately before running that code. Validating code immediately before its use can reduce the risk of TOCTOU attacks (Time of Check to Time of Use). • At each stage of the boot sequence, wherever possible, check that only the expected hardware is present and matches the stage's configuration parameters. • Do not boot the next stage of device functionality until the previous stage has been successfully booted. • Ensure failures at any stage of the boot sequence fail gracefully into a secure state, to ensure no unauthorised access is gained to underlying systems, code or data (for example, via a U-Boot prompt). Any code run must have been previously authenticated.

Aspect	Risk controls
Secure operating system	A majority of IoT devices will use open-source software and therefore applying user accounts maybe restricted to local access only. • Include in the operating system (OS) only those components (libraries, modules, packages etc.) that are required to support the functions of the device. • Shipment should include the latest stable OS component versions available. • Devices should be designed and shipped with the most secure configuration in place. A decision to reduce security must be a justified and documented decision made downstream from shipment if necessary. • Ensure the OS is securely booted. • Update OS components to the latest stable versions throughout the lifecycle of a deployed device. • Disable all unsecure and unused ports, protocols and services. • Set permissions so users/applications cannot write to the root file system. • Accounts for ordinary users/applications must have least privilege access to perform the necessary functions. Separate administrator accounts (if required) will have greater rights of access. Do not run anything as root unless genuinely unavoidable. • Apply access controls and least privilege to files and data and rights to perform the required functions. • Implement an encrypted file system. • Document the security configuration of the OS. • Use Change Control management to apply changes to the OS.

Aspect	Risk controls
Application Security considerations for procurement /Risk assessment	• Listed below are points to raise when procuring commercial off the shelf or bespoke software during a risk assessment. • Document the security design of applications. • Applications must be operated at the lowest privilege level possible, not as root. • Applications must only have access to those resources they need • Applications should be isolated from each other. For example, use sandboxing techniques such as virtual machines, containerisation. • Ensure compliance with UK Police data processing. • Incorporate security into all stages of the software development lifecycle, including software design, secure source code storage and traceability, code reviews, code analysis tools etc • Use secure design and coding techniques. For example, sanitise and validate all data input before processing, prevent buffer overruns, use secure protocols and remove weak encryption ciphers. • Ensure all errors are handled gracefully and any messages produced do not reveal any sensitive information. • Never hard-code credentials into an application. Credentials must be stored separately in secure trusted storage and must be updateable in a way that ensures security is maintained. • Remove all default user accounts and passwords. • Ensure 3rd party application software and libraries, whether commercial off-the-shelf COTS or specifically bespoke developed, follow these security guidelines wherever possible. • Use the most recent stable version of the operating system and libraries. • Ensure applications are within the scope of vulnerability and patch management processes. • Never deploy debug versions of code. The distribution should not include compilers, files containing developer comments, sample code, or other superfluous files. • Consider the impact off Availability of the application/system if network connectivity is lost. maintain functionality and security wherever possible

Aspect	Risk controls
Credential Management	• A device should be uniquely identifiable by means of a factory-set tamper resistant hardware identifier if possible. • Apply the NSCP Password policy • Each password stored for authenticating credentials must use an industry standard hash function, along with a unique salt value that is not obvious (for example, not a username). • Passwords stored for use as credentials must be strongly encrypted, using an industry standard algorithm. • Store credentials or encryption keys in a Secure Access Module (SAM), Trusted Platform Module (TPM), Hardware Security Module (HSM) or trusted key store if possible. • Use MFA for accessing sensitive data. • Ensure a trusted & reliable time source is available where authentication methods require this, e.g. for digital certificates. • Digital certificates require careful management as part of an effective credential solution. A certificate has a defined lifetime (and may need to be revoked) so should not be just deployed and then forgotten. Further discussion on certificates and their management is available at the link below*. • There must be a secure reliable means to update a digital certificate and its certificate chain on a device before it expires. • Certificate used to identify a device must be unique and only used to identify that one device. Do not reuse the certificate across multiple devices. • A 'factory reset' function must fully remove all user data/credentials stored on a device and apply data destruction as per NIST guidelines
Encryption	• **See NCSP Cryptography standard** • Use industry-standard cypher suites, use the strongest algorithms and always use the most recent version of an encryption protocol. • When configuring a secure connection, if an encryption protocol offers a negotiable selection of algorithms, remove weaker options so they cannot be selected for use in a downgrade attack. • Store encryption keys in a Secure Access Module (SAM), Trusted Platform Module (TPM), Hardware Security Module (HSM) or trusted key store if possible. • Do not use insecure protocols, e.g. FTP, Telnet. • It should be possible to securely replace encryption keys remotely. • If implementing public/private key cryptography, use unique keys per device and avoid using global keys. A device's private key should be generated by that device or supplied by an associated secure credential solution, e.g. smart card. It

Aspect	Risk controls
	should remain on that device (or associated solution) and never be shared/visible to elsewhere. Conversely, a device's public key may be shared elsewhere to support encryption with this device.
Network Connections	• Activate only those network interfaces that are required (wired, wireless - including Bluetooth etc.). • Run only those services on the network that are required. • Only open network ports that are required. • Apply a software firewall on the device if possible. • Always use secure protocols, e.g. HTTPS, SFTP. • Never exchange credentials in clear text or over weak solutions such as HTTP Basic Authentication. • Authenticate every incoming connection to ensure it comes from a legitimate source. • Authenticate the destination before sending sensitive data. • Consider segregating IoT devices onto a separate network or virtual networks.
Securing software updates	• Encrypt update packages to hinder reverse engineering. • The update routine must cryptographically validate the integrity and authenticity of a software update package before installation begins. Updates performed during a device re-boot must include these checks as part of the secure boot process - see Best Practice Guide • Device Secure Boot. • Ensure that the package cannot be modified or replaced by an attacker between being validated and installed - a TOCTOU (Time of Check to Time of Use) attack. • To ensure a complete set of security fixes is applied during an update, the installation routine must automatically determine all required dependencies and install any previous versions of the software as required. If unable to resolve all dependencies, the device must be left in a safe, functioning state and details made available to the installer. • A 'fail safe' mechanism is required that will leave a device in a known safe state in the event an update fails. • Implement a trusted anti-rollback function, to prevent unauthorised reversion to earlier software versions with known security vulnerabilities.

Aspect	Risk controls
Logging	• Ensure all logged data comply with prevailing data protection regulations. • Run the logging function in its own operating system process, separate from other functions. • Store log files in their own partition, separate from other system files. • Set log file maximum size and rotate logs. • Where logging capacity is limited, ensure that logging includes start-up and shutdown parameters, login/access attempts and anything unexpected. • Restrict access rights to log files to the minimum required to function. • If logging to a central repository, send log data over a secure channel if the logs carry sensitive data and/or protection against tampering of logs must be assured. • Implement log 'levels' so that lightweight logging can be the standard approach, but with the option to run more detailed logging when required. • Monitor and analyse logs regularly to extract valuable information and insight. • Synchronise to an accurate time source, where possible, so log file time stamps can be easily correlated. • Passwords and other secret information should not ever be displayed in logs. **Contact the NMC for support and guidance on logging and monitoring**

Aspect	Risk controls
Software update policy	• Management of all connected devices over their complete device lifecycle. • Maintain an inventory of devices integrated into a force's network. • Actively maintaining version information about software deployed on devices. • Processes for planned device updating and rapid deployment of critical updates. • Identification of unfixable or non-updateable devices that have known attack vectors, and processes to ensure that such devices are prevented from compromising the security of the system, for example through device revocation or some other reliable method. • Securely managing change of device ownership. • Securely managing devices at their end of life. • A clear, publicised, process for managing software errata. This process must enable developers, users and security researchers to report security vulnerabilities and other issues and must enable rapid communication to users. It should also: • Define a process for identifying affected configurations. • Define the circumstances that require a software update to be developed and released. • Define the urgency of releasing an update, based on the potential impact of the threat to the user, vendor and other users of the network, and impact to the user of deploying the update. • Define the procedure for updating software on devices. • Identify clear ownership and escalation processes within the organisation. Mechanisms for software updates must be clearly defined within the software architecture. • The policy must recognise that existing standards for software patching, such as NIST SP800-40, may well need to be adapted for updating software on IoT systems

Aspect	Risk controls
Assessing secure Boot Process	• The secure boot cannot be bypassed. • All code loaded as part of the boot process, unless it runs legacy hardware directly from ROM , is verified to ensure: a. The code was created by the expected, authorised source b. The code has not been modified since it was created c. The code is intended for the device type on which it is to be run d. Verify code after it has been loaded into RAM (this is preferable to verifying it 'at rest' in storage). • The boot sequence starts running from ROM, using an immutable root key to verify the first code to be loaded. (Consider multiple immutable root keys for verifying different boot stages, for generating derived keys, or even for redundancy in case of subsequent compromise). • Modules of code are loaded progressively, but only after each previous stage has been successfully verified and booted. • Any existing data currently installed on the device that will be used as part of the boot configuration is checked for length, type, range etc. prior to use within the boot process. • At each stage of the boot process, wherever possible, the boot software checks that the hardware configuration matches the expected configuration parameters for that stage. • The boot process ensures that if an error occurs during any stage of the process, the device "fails securely" i.e. into a secure state in which RAM has been cleared of residual code. Secure failure must also ensure the device is not 'bricked' and no unauthorised access can be made to underlying systems, code or data (e.g. via a U-Boot prompt). • The manufacturer implements a secure process for generating keys and certificates. The provisioning, storage and usage of keys and certificates on devices is secure. Device end-of-life management ensures the security of keys and certificates is maintained. • Bootloader code is updated to address vulnerabilities (although this may not be possible in the initial ROM stage)

Aspect	Risk controls
Software image and update signing	System builder should ensure the following: • The signature's cryptographic key size and hash algorithms have sufficient cryptographic strength for the intended service life of the product. That the signature method chosen has a key provisioning method suitable for the intended manufacturing supply chain (Be mindful there may be a cause to replace keys at some point). • The system makes use of any hardware cryptography support available on the device, such as hardware key store, accelerated hashing and decryption operations. • The copies of the symmetric or asymmetric keys used to create the software component signatures are stored in sufficiently secure storage Hardware Security Module (HSM)compliant with FIPS-140.Please refer to the Key management Standards
Side channel attacks	• Evaluate the risks to determine the level of protection needed and the impacted modules. • Obfuscate signals by varying amplitude and/or time domain. Randomise jitter and delay. • Obfuscate hardware and software-based functions with randomised performance (or constant performance) regardless of the inputs. • Insert dummy data operations. • Mask cryptographic functions and/or employ dedicated cryptographic modules. • Design circuitry to fail gracefully and reliably as power supply rails reach designed limits. • Design circuitry to fail gracefully and reliably as temperature reaches designed limits. • Implement error checking. • Include fault injection countermeasures in the design. • Employ appropriate mitigating physical construction (mountings, housing, EMF shields etc.).

Electric Vehicles / Connected Vehicles

With the use of electric / connected vehicles being deployed in police forces it is important to understand that like any other IoT device an electric vehicle must be viewed in same aspect due to the components used.

By following the vehicle security lifecycle this would help identify any potential security risks in electric vehicles, the vehicle security lifecycle is a series of key headers supported with a list of questions to ask when reviewing the security of an electric vehicle.

Procurement

- What data is collected about the vehicle and sent to the manufacturer(telematics)?
- What smart features are present on the vehicle?
- Does the vehicle support over – the- air updates?
- Is the country of origin considered?
- Has the specification been minimised with manufacturer?
- Has the infotainment system been wiped if purchased second hand?

Commissioning /Onboarding

- Is there an understanding of what is connected to the diagnostic ports such as OBD?
- To avoid unauthorised connectivity of vehicle data review if there are any remote diagnostics.
- Locate all the USB ports and insert with USB Blockers to prevent unauthorised USB connectivity
- Are there any unexplained or unexpected trackers present?
- Ensue that the SIMS (Subscriber Identity Module) is present in the vehicle to enable networking and security features.
- The infotainment system should be configured to minimise any risks associated with the communications and data stored in the vehicle, disable unnecessary features and personal data to prevent unauthorised access and data breaches.
- To prevent fraud and theft Ensure that the VIN (Vehicle Identification Number) has been obscured, this will also prevent vehicle cloning and identity theft.

Maintenance

- If a third-party telematics or fleet management company telematics is being used ensure that the systems have undergone supplier assurance e.g. Third-Party Assurance for Policing (TPAP) . (The IT systems or data storage may not have had a highly secure law enforcement use case in place during the design phase.)
  Validate if the vehicle configuration changed since it was commissioned?

- How are electric / connected vehicles maintained?
- Conduct third-party security assessments of new charging installations to ensure data and vehicles are protected.
- Ensure that the vehicle’s MOT is carried out securely.
- Understood and mitigated the risks posed by vehicle diagnostic tooling.

End of Life

- Remove any external storage devices from the vehicle.
- Reset the infotainment system.
- Delete Bluetooth, phone, contact and GPS information and signed out of apps.
- Remove any additional telematics systems or trackers that have been fitted to the vehicle.
- In particularly sensitive cases, remove and destroy the infotainment system prior to a vehicle being scrapped or be crushed.

For further guidance refer to the National Protective Security Authority guidance

- A guide for law enforcement procurement, maintenance and management
- Managing vehicle digital footprints

- The IoT Security Foundation

- NIST IoT Device Cybersecurity Guidance
- ENISA IoT Security Recommendations
- The Internet Society’s Online Trust Alliance
- OWASP IoT Top Ten
- ISO/IEC 27001 for Information Security Management

Communication approach

This document will be communicated as follows:

- Internal peer review by the members of the National Cyber Policy & Standards Working Group (NCPSWG), which includes PDS and representatives from participating forces.
- Formal publication and external distribution to PDS community, police forces and associated bodies.

Measurables generated by adopting this guideline can also form part of regular cyber management reporting.

Review Cycle

This guideline will be reviewed at least annually (from the date of publication) and following any major change to Information Assurance (IA) strategy, membership of the community, or an identified major change to the cyber threat landscape. This ensures IA requirements are reviewed and that the guideline continues to meet the objectives and strategies of the police service.

Document Compliance Requirements

(Adapt according to Force or PDS Policy needs.)

Equality Impact Assessment

(Adapt according to Force or PDS Policy needs.)

Document Information

Document Location

https://knowledgehub.group/web/national-standards/policing-standards

Revision History

Version	Author	Description	Date
0.2	PDS Cyber	Final draft for NCPSWG review	06/06/25

Approvals

Version	Name	Role	Date
1.0	NCPSWG	National Cyber Policy & Standards Working Group	20/08/25

Document References

Document Name	Version	Date
ISF - Standard of Good Practice (for Information Security)	v2024	03/2024
ISO 27002:2022 - Information security, Cybersecurity and privacy protection – Information security controls	v2022	02/2022
CIS Controls	v8	05/2021
NIST Cyber Security Framework	v1.1	04/2018
CSA Cloud Controls Matrix	v4	01/2021
10 Steps to Cyber Security - NCSC.GOV.UK	Web Page	05/2021