Checklist for Auditors/Assessors

(STQC/IoTSCS/F04)

Issue :01

Checklist for Auditors/Assessors

Cl.No	Requirements as per 'ISO/IEC 27402 IoT security and privacy — Device baseline requirements	Compliance Status			Observation
		Yes	No	N/A	
5.1	**Requirements for IoT device policies and documentation**				
5.1.1	**Risk management**				
5.1.1.1.1	IoT devices shall have documentation recording the results of a risk assessment process performed at the IoT device level in the context of a risk assessment at the system level.				
5.1.1.1.2	The risk assessment process shall take into account intended outcomes for the intended use case.				
5.1.1.1.3	The risk assessment process shall also take into account the needs and expectations of interested parties (e.g. those parties on networks to which the IoT device is connected), including physical and logical undesired effects.				
5.1.1.1.4	The risk assessment shall take into account that IoT devices can be constrained (e.g. limited battery, little memory, 'weak' CPU), which informs the risk treatment process.				
5.1.1.1.5	Risk assessment and treatment processes shall be defined and applied as follows:				
	a) determine if separate risk assessment and treatment processes are necessary for different products;				
	b) select appropriate risk treatment options, taking account of the risk assessment results;				
	c) determine all controls that are necessary to implement the risk treatment option(s) chosen;				
	d) identify all security and privacy features of the IoT device from the controls identified in c) above;				
	e) compare the features identified in d) above with those in 5.2, and verify that no necessary features have been omitted;				
	f) produce a Statement of Applicability that contains the necessary features [see steps d) and e)] and justification for inclusions and the justification for exclusions of features from 5.2;				

	g) if other standards related to device requirements are used, implement the requirements of those standards after steps a) through to f);				
	h) formulate a risk treatment plan;				
	i) inform the risk owner of the risk treatment plan and any residual risks, or where applicable, obtain their approval of the plan and acceptance of the residual risks.				
5.1.1.1.6	IoT devices shall implement the features and controls identified as necessary in its Statement of Applicability, as well as features and controls identified in 5.1.1.1.5, step g).				
5.1.1.1.7	The documentation shall be available for the supported lifetime of the product.				
5.1.2	**Information disclosure**				
5.1.2.1.1	IoT devices shall have user documentation that lists the features that the IoT device provides to support controls for security and privacy, making it clear if any of the IoT device requirements in 5.2 are not included.				
5.1.2.1.2	Such information shall be publicly available for the period of time the IoT device is supported.				
5.1.2.1.3	IoT devices shall be covered by a security support policy and other supporting documentation wherein users are made aware in advance of when security updates will be discontinued.				
5.1.3	**Vulnerability disclosure and handling processes**				
5.1.3.1.1	IoT devices shall have documentation that defines the vulnerability disclosure and handling processes that will apply for the supported lifetime of the device.				
5.1.3.1.2	Vulnerability disclosure and handling processes shall include, at a minimum, a capability to receive reports of potential vulnerabilities from the public.				
5.2	**Requirements for IoT device capabilities and operations**				
5.2.1	**General- This clause includes IoT device features to be used with a risk assessment and treatment process in accordance with 5.1.1.**				

5.2.2	**Configuration**				
5.2.2.1.1	If the configuration settings of the IoT device can be modified, only authorized entities shall be able to modify the configuration settings of the IoT device.				
5.2.2.1.2	If IoT devices are capable of changing the configuration of IoT and other devices, they shall only be capable of making such changes when authorized.				
5.2.3	**Software reset**				
5.2.3.1.1	If IoT devices have the capability to be reset, that process shall be secure.				
5.2.3.1.2	This capability shall only be executable by an authorized entity.				
5.2.4	**User data removal**				
5.2.4.1.1	If the IoT device stores user data, it shall provide a function for deleting appropriate user data stored on the device in any type of memory.				
5.2.4.1.2	The function shall be restricted to authorized entities only.				
5.2.5	**Protection of data**				
5.2.5.1.1	IoT devices shall be capable of protecting the data they store and transmit from unauthorized access, modification and disclosure.				
5.2.5.1.2	This shall include configuration settings, identifying data, user data, event logs and sensitive security parameters.				
5.2.5.1.3	IoT devices shall be capable of protecting their software (including firmware) from unauthorized access and modification.				
5.2.5.1.4	IoT devices shall use cryptography (e.g. encryption with authentication, cryptographic hashes, digital signature validation) to prevent the confidentiality and integrity of data requiring protection from being compromised.				
5.2.5.2	**Additional recommendation**				
5.2.5.2.1	**General** When IoT devices are started up, they should check the integrity and authenticity of the software and/or firmware and enforce security controls. If the IoT				

	device fails these checks, it should: — notify the user of any violation, — render itself inoperable, — operate in a fail-safe mode that provides security protection, or — initiate device recovery if recovery actions can be performed with integrity.				
	Upon first installation or maintenance, IoT devices should set themselves to secure default configurations. User configuration options should prevent users from choosing insecure configurations or provide a warning. If capable, IoT devices should have the ability to provide compartmentalization. IoT devices should use function modules to restrict access to system resources, which should only be granted to authorized entities. Trusted computing bases (TCB) should be kept as small as possible to minimize the surface that is exposed to attackers and to reduce the probability that a bug or feature can be used to circumvent security protections. Memory protection mechanisms such as memory safe languages, stack canaries, address space layout randomization (ASLR) and limited or no execute permissions are recommended wherever applicable.				
5.2.5.2.2	**Event logging** If capable, IoT devices should record sufficient details for each event to facilitate an authorized entity's ability to identify anomalous events and meaningfully analyse the associated data.				
5.2.5.2.3	**Sensitive security parameters** The outcome of the risk assessment in 5.1.1 should help determine whether an IoT device may include hard-coded or shared sensitive security parameters, if such parameters are unique per device and not universal.				
5.2.5.3	**Additional information**				
5.2.5.3.1	**General** Hardware-based solutions such as built-in crypto accelerators and dedicated hardware can enhance the use of cryptographic modules and cryptographic key protection capabilities to				

	protect the data in storage and transit to meet the performance requirements. Physical countermeasures can support resistance to side channel attacks. Such functions can include hardware-based root of trust (RoT). RoT is a foundational feature to provide platform integrity and ensure a foundation to develop and support the device's chain of trust. The root of trust is ideally based on a hardware-validated boot process to ensure the system can be started using code from an immutable source. As such, RoT is essential to enable platform attestation including for a verified boot process. When used to protect secrets and device correctness, hardware can support a foundational root of trust upon which rich software functionality can be implemented more securely and safely. Compartments are protected by hardware-enforced boundaries to prevent a flaw or breach in one software compartment from propagating to other software compartments in the system. Compartmentalization introduces additional protection boundaries within the hardware and software stack to create additional layers of defence in depth. For example, a common technique is to use operating systems processes or independent virtual machines as compartments. Integrity checking and recovery modes may not be appropriate in safety critical applications where continuous operation is essential.				
5.2.5.3.2	**Event logging** Implementation of event logging, including editing of logs, depends on device storage capabilities. IoT devices can support remote logging.				
5.2.6	**Interface access**				
5.2.6.1.1	IoT devices shall have mechanisms to limit logical access to its interfaces to authorized entities only.				
5.2.6.1.2	IoT devices shall employ appropriate authentication and access control mechanisms.				
5.2.6.1.3	Security and privacy requirements shall be assessed when designing and implementing the functions of				

	IoT devices regarding creation and use of identifiers.				
5.2.6.1.4	IoT devices shall ensure that common values for critical security parameters, such as global private keys or standard passwords, are replaced by values that are unique per device or explicitly defined by an appropriate external entity before they are put into operation.				
5.2.6.2	**Additional recommendation(s)** The IoT device should be capable of being logically identified. While identifiers can enable a host of cybersecurity controls (such as asset management, automatic device discovery, and software updates), creating or using persistent identifiers should be avoided unless such use is unavoidable. Where such uses arise, the existence of such identifiers should be made clear to users. Mechanisms to limit logical access (to authorized entities) should be applied to the following: a) the ability to enable or disable, through software or hardware means, any interfaces (including local and network interfaces); b) the ability to restrict access (e.g. through authentication) to all remote interfaces; c) the ability to identify or block devices not supported by an IoT device when it is attempting to access interfaces.				
5.2.6.3	**Additional information**				
5.2.6.3.1	**General** Examples of user interfaces include administrative consoles, web pages, APIs or other externally-exposed IoT device interfaces. Injection, XML external entities, cross site scripting and insecure deserialization are examples of common attacks to remote interfaces. Hardware-based capabilities can harden interface access protection against privilege escalation and control-flow attacks.				
5.2.6.3.2	**Identifiers**				

	IoT devices can use identifiers in order to operate within an IoT system. Examples of such identifiers include serial numbers, cryptographic keys, and certificates.				
5.2.7	**Software and firmware updates**				
5.2.7.1.1	If the IoT device supports software updates, updates shall be performed using a secure procedure.				
5.2.7.1.2	Updates shall only be initiated by authorized entities.				
5.2.7.1.3	Unexpected interruption of an update shall leave the IoT device in a state that minimizes potential for harm, taking into account the risks of the IoT device not functioning as expected.				
5.2.8	**User Notifications**				
	IoT devices to notify users about about a negative event or condition. Some IoT devices do not have capabilities to actively inform the user (e.g. write a message on the screen, emit a sound or light), but they can respond with a message when queried or accessed remotely. IoT devices that do not have capabilities to directly inform users can send notifications and alerts via a local hub. A user query can be as simple as trying to access the device with a browser, mobile application, or something more complex. Alternatively, IoT devices can send a message to an alarm, monitoring, or logging device within the IoT system.				

IoTVS Level 1

IoTVS Level 1 requirements aim to provide a security baseline for connected devices which does not allow an attacker to move laterally to other devices or systems on the IoT ecosystem.

#	Description	Mapping with Requirements as per 'ISO/IEC 27402	Observation
C.1	Verify that application layer debugging interfaces such USB, UART, and other serial variants are disabled or protected by a complex password.	Interface access	
C.2	Verify that cryptographic keys and certificates are unique to each individual device.	Interface access	
C.3	Verify that memory protection controls such as ASLR and DEP are enabled by the embedded/IoT operating system, if applicable.	Protection of data	
C.4	Verify that on-chip debugging interfaces such as JTAG or SWD are disabled or that available protection mechanism is enabled and configured appropriately.	Interface access	
C.5	Verify that trusted execution is implemented and enabled, if available on the device SoC or CPU.	Protection of data	
C.6	Verify that sensitive data, private keys and certificates are stored securely in a Secure Element, TPM, TEE (Trusted Execution Environment), or protected using strong cryptography.	Protection of data	
C.7	Verify that the firmware apps protect data-in-transit using transport layer security.	Protection of data	
C.8	Verify that the firmware apps validate the digital signature of server connections.	Protection of data	
C.9	Verify that wireless communications are mutually authenticated.	Protection of data	
C.10	Verify that wireless communications are sent over an encrypted channel.	Protection of data	
C.11	Verify that any use of banned C functions are replaced with the appropriate safe equivalent functions.	General	
C.12	Verify that each firmware maintains a software bill of materials cataloging third-party components, versioning, and published vulnerabilities.	Vulnerability disclosure and handling processes	

C.13	Verify all code including third-party binaries, libraries, frameworks are reviewed for hardcoded credentials (backdoors).	General	
C.14	Verify that the application and firmware components are not susceptible to OS Command Injection by invoking shell command wrappers, scripts, or that security controls prevent OS Command Injection.	General	

Checklist for Auditors/Assessors

IoTVS Level 2

IoTVS Level 2 is for IoT devices that contain sensitive data, which requires protection and is the recommended level for most devices.

#	Description	Mapping with Requirements as per 'ISO/IEC 27402	Observation
C.1	Verify that application layer debugging interfaces such USB, UART, and other serial variants are disabled or protected by a complex password.	Interface access	
C.2	Verify that cryptographic keys and certificates are unique to each individual device.	Interface access	
C.3	Verify that memory protection controls such as ASLR and DEP are enabled by the embedded/IoT operating system, if applicable.	Protection of data	
C.4	Verify that on-chip debugging interfaces such as JTAG or SWD are disabled or that available protection mechanism is enabled and configured appropriately.	Interface access	
C.5	Verify that trusted execution is implemented and enabled, if available on the device SoC or CPU.	Protection of data	
C.6	Verify that sensitive data, private keys and certificates are stored securely in a Secure Element, TPM, TEE (Trusted Execution Environment), or protected using strong cryptography.	Protection of data	
C.7	Verify that the firmware apps protect data-in-transit using transport layer security.	Protection of data	
C.8	Verify that the firmware apps validate the digital signature of server connections.	Protection of data	
C.9	Verify that wireless communications are mutually authenticated.	Protection of data	
C.10	Verify that wireless communications are sent over an encrypted channel.	Protection of data	
C.11	Verify that any use of banned C functions are replaced with the appropriate safe equivalent functions.	General	
C.12	Verify that each firmware maintains a	Vulnerability disclosure and	

	software bill of materials cataloging third-party components, versioning, and published vulnerabilities.	handling processes	
C.13	Verify all code including third-party binaries, libraries, frameworks are reviewed for hardcoded credentials (backdoors).	General	
C.14	Verify that the application and firmware components are not susceptible to OS Command Injection by invoking shell command wrappers, scripts, or that security controls prevent OS Command Injection.	General	
C.15	Verify that the firmware apps pin the digital signature to a trusted server(s).	General	
C.16	Verify the presence of tamper resistance and/or tamper detection features.	General	
C.17	Verify that any available Intellectual Property protection technologies provided by the chip manufacturer are enabled.	General	
C.18	Verify security controls are in place to hinder firmware reverse engineering (e.g., removal of verbose debugging symbols).	General	
C.19	Verify the device validates the boot image signature before loading.	General	
C.20	Verify that the firmware update process is not vulnerable to time-of-check vs time-of-use attacks.	Software and firmware updates	
C.21	Verify the device uses code signing and validates firmware upgrade files before installing.	Software and firmware updates	
C.22	Verify that the device cannot be downgraded to old versions (anti-rollback) of valid firmware.	Software and firmware updates	
C.23	Verify usage of cryptographically secure pseudo-random number generator on embedded device (e.g., using chip-provided random number generators).	Protection of data	
C.24	Verify that firmware can perform automatic firmware updates upon a predefined schedule.	Software and firmware updates	

IoTVS Level 3

IoTVS Level 3 is for the most critical IoT devices that perform high value transactions, contain sensitive medical data, or any application that requires the highest level of trust.

#	Description	Mapping with Requirements as per 'ISO/IEC 27402	Observation
C.1	Verify that application layer debugging interfaces such USB, UART, and other serial variants are disabled or protected by a complex password.	Interface access	
C.2	Verify that cryptographic keys and certificates are unique to each individual device.	Interface access	
C.3	Verify that memory protection controls such as ASLR and DEP are enabled by the embedded/IoT operating system, if applicable.	Protection of data	
C.4	Verify that on-chip debugging interfaces such as JTAG or SWD are disabled or that available protection mechanism is enabled and configured appropriately.	Interface access	
C.5	Verify that trusted execution is implemented and enabled, if available on the device SoC or CPU.	Protection of data	
C.6	Verify that sensitive data, private keys and certificates are stored securely in a Secure Element, TPM, TEE (Trusted Execution Environment), or protected using strong cryptography.	Protection of data	
C.7	Verify that the firmware apps protect data-in-transit using transport layer security.	Protection of data	
C.8	Verify that the firmware apps validate the digital signature of server connections.	Protection of data	
C.9	Verify that wireless communications are mutually authenticated.	Protection of data	
C.10	Verify that wireless communications are sent over an encrypted channel.	Protection of data	
C.11	Verify that any use of banned C functions are replaced with the appropriate safe equivalent functions.	General	
C.12	Verify that each firmware maintains a software bill of materials cataloging third-party components, versioning, and published vulnerabilities.	Vulnerability disclosure and handling processes	
C.13	Verify all code including third-party binaries, libraries, frameworks are reviewed for hardcoded credentials	General	

	(backdoors).		
C.14	Verify that the application and firmware components are not susceptible to OS Command Injection by invoking shell command wrappers, scripts, or that security controls prevent OS Command Injection.	General	
C.15	Verify that the firmware apps pin the digital signature to a trusted server(s).	General	
C.16	Verify the presence of tamper resistance and/or tamper detection features.	General	
C.17	Verify that any available Intellectual Property protection technologies provided by the chip manufacturer are enabled.	General	
C.18	Verify security controls are in place to hinder firmware reverse engineering (e.g., removal of verbose debugging symbols).	General	
C.19	Verify the device validates the boot image signature before loading.	General	
C.20	Verify that the firmware update process is not vulnerable to time-of-check vs time-of-use attacks.	Software and firmware updates	
C.21	Verify the device uses code signing and validates firmware upgrade files before installing.	Software and firmware updates	
C.22	Verify that the device cannot be downgraded to old versions (anti-rollback) of valid firmware.	Software and firmware updates	
C.23	Verify usage of cryptographically secure pseudo-random number generator on embedded device (e.g., using chip-provided random number generators).	Protection of data	
C.24	Verify that firmware can perform automatic firmware updates upon a predefined schedule.	Software and firmware updates	
C.25	Verify that the device wipes firmware and sensitive data upon detection of tampering or receipt of invalid message.	User data removal	
C.26	Verify that only micro controllers that support disabling debugging interfaces (e.g. JTAG, SWD) are used.	Interface access	
C.27	Verify that only micro controllers that provide substantial protection from de-capping and side channel attacks are used.	Protection of data	
C.28	Verify that sensitive traces are not exposed to outer layers of the printed circuit board.	Interface access	
C.29	Verify that inter-chip communication is encrypted (e.g. Main board to daughter board communication).	Protection of data	

C.30	Verify the device uses code signing and validates code before execution.	Protection of data	
C.31	Verify that sensitive information maintained in memory is overwritten with zeros as soon as it is no longer required.	User data removal	
C.32	Verify that the firmware apps utilize kernel containers for isolation between apps.	General	
C.33	Verify that secure compiler flags such as -fPIE, -fstack-protector-all, -Wl,-z,noexecstack, -Wl,-z,noexecheap are configured for firmware builds.	General	
C.34	Verify that micro controllers are configured with code protection (if applicable).	Protection of data	

Controls List for ISO IEC 27400

To audit IoT Systems

Sr. No.	Control	Objective	Applicability	Observation
7.1.2	**Security controls for IoT service developer and IoT service provider**			
7.1.2.1	**Policy for IoT security**			
Control 01	A policy for IoT security should be defined, approved by management, published, communicated to relevant personnel and relevant external parties and reviewed at planned intervals or if significant changes occur.	To provide management direction and support for IoT security within the IoT service developer or the IoT service provider in accordance with business requirements, expectations of stakeholders and relevant laws and regulations.	IoT service developer/ IoT service provider	
7.1.2.2	**Organization of IoT security**			
Control 02	Roles and responsibilities for security of IoT should be defined and allocated.	To establish and maintain a management framework to initiate and control the implementation and operation of IoT security within the IoT service provider or the IoT service developer.	IoT service developer/ IoT service provider	
7.1.2.3	**Asset management**			

Control 03	Information, IoT devices and systems and their functions and operations to be protected should be identified	To identify assets of IoT devices and systems for designing appropriate protecting measures	IoT service provider	
7.1.2.4	**Equipment and assets located outside physical secured areas**			
Control 04	Specific security measures should be applied to IoT equipment and assets which are located or operated outside physical secured areas.	To prevent loss, damage, theft or compromise of IoT devices and interruption to the operation of IoT services.	IoT service provider	
7.1.2.5	**Secure disposal or re-use of equipment**			
Control 05	All items of equipment containing storage media should be verified to ensure that any sensitive data and licensed software has been removed or securely overwritten prior to disposal or re-use.	To prevent information leakage and malicious use of the IoT device and other equipment of the IoT system at its disposal or re-use.	IoT service provider	
7.1.2.6	**Learning from security incidents**			
Control 06	Knowledge gained from analyzing and resolving IoT security incidents should be used to reduce the likelihood or impact of future incidents.	To reduce negative effects of incidents in the provision and use of IoT services.	IoT service developer/ IoT service provider	
7.1.2.7	**Secure IoT system engineering principles**			
Control 07	Principles for engineering secure IoT systems that address designing and implementation of security functions defense in depth and hardening of systems and software should be applied to the development of IoT systems.	To ensure that security is designed and implemented in the development of IoT systems.	IoT service developer	
7.1.2.8	**Secure development environment and procedures**			
Control 08	Secure development environment and procedures should be applied to the development of IoT systems.	To avoid introduction of insecurity to IoT systems during development.	IoT service developer	
7.1.2.9	**Security of IoT systems in support of safety**			
Control 09	Security principles in support of safety should be applied to the development of IoT systems.	To support safety in IoT systems.	IoT service developer/ IoT service	

			provider	
7.1.2.10	**Security in connecting varied IoT devices**			
Control 10	An IoT system should be designed and implemented to ensure and maintain security in connecting varied IoT devices.	To maintain security of IoT system in connecting varied IoT devices including those not necessarily verified by the IoT service developer or the IoT service provider.	IoT service developer/ IoT service provider	
7.1.2.11	**Verification of IoT devices and systems design**			
Control 11	Design and implementation of IoT devices and IoT systems should be verified.	To ensure security and safety of the IoT device and IoT system.	IoT service developer/ IoT service provider	
7.1.2.12	**Monitoring and logging**			
Control 12	States, events and network traffics of IoT devices and systems should be monitored and logged.	To detect and trace abnormalities and incidents of IoT devices and systems.	IoT service developer/ IoT service provider	
7.1.2.13	**Protection of logs**			
Control 13	Logs for IoT devices and systems should be protected from leakage, destruction and unintended alteration.	To ensure the capability and reliability of logging.	IoT service developer/ IoT service provider	
7.1.2.14	**Use of suitable networks for the IoT systems**			
Control 14	Applied network and communication technologies for IoT and systems should meet the needs of communication function, capacity and security, and of function and performance of IoT devices.	To use the network that meets security, performance and other needs of the IoT system.	IoT service developer/ IoT service provider	
7.1.2.15	**Secure settings and configurations in delivery of IoT devices and services**			
Control 15	IoT devices and services should be delivered with secure settings and configurations.	To ensure security of IoT devices and services in delivery.	IoT service developer/ IoT service provider	
7.1.2.16	**User authentication**			
Control 16	Authentication function of users and IoT devices for accessing IoT systems and services should be implemented and applied.	To protect information, IoT devices, systems and services from unauthorized access and other security breaches.	IoT service developer/ IoT service provider	

7.1.2.17	**Provision of software and firmware updates**			
Control 17	Mechanism for updating software and firmware of IoT devices and systems should be designed, implemented and operated.	To ensure security for updating software and firmware of IoT device and IoT system.	IoT service developer/ IoT service provider	
7.1.2.18	**Sharing vulnerability information**			
Control 18	Vulnerabilities of IoT devices, systems and services should be monitored and informed to the IoT users and relevant parties along with associated risks.	To ensure relevant stakeholders are informed of vulnerabilities of IoT devices, systems and services and aware of derived risks.	IoT service developer/ IoT service provider	
7.1.2.19	**Security measures adapted to the life cycle of IoT system and services**			
Control 19	Security measures of the IoT system and service should be adapted to and kept during the stages of the life cycle, including their development, operation, maintenance and destruction.	To maintain security of IoT system and service throughout the life cycle.	IoT service developer/ IoT service provider	
7.1.2.20	**Guidance for IoT users on the proper use of IoT devices and services**			
Control 20	The IoT users should be provided with guidance on the proper use of IoT devices with risks and undesirable effects of IoT system and service that can be derived from improper use of IoT devices.	To make the IoT users aware of the security risks in the use of IoT devices, and to ensure implementation of security measures.	IoT service developer/ IoT service provider	
7.1.2.21	**Determination of security roles for stakeholders**			
Control 21	Roles of IoT service developer, IoT service provider and other stakeholders in security of IoT system and service should be determined and agreed among relevant parties.	To ensure security of IoT system and service that involves entities participating in the provision and use of IoT system and service.	IoT service developer/ IoT service provider	
7.1.2.22	**Management of vulnerable devices**			
Control 22	Vulnerable IoT devices should be detected recorded, and alerts provided to IoT users and administrators of these devices.	To maintain IoT devices to be secure.	IoT service provider	
7.1.2.23	**Management of supplier relationships in IoT security**			
Control 23	Specifications and supporting obligations of suppliers for information security of IoT device	To ensure continued provision of secure IoT device and service.	IoT service developer/ IoT service	

	and IoT service should be managed by the acquiring organization based on the contracts with suppliers.		provider	
7.1.2.24	**Information security in IoT devices**			
Control 24	Information security controls of IoT devices should be documented and only disclosed to the parties that require them.	-	IoT service developer	
7.1.3	**Security controls for IoT Users**			
7.1.3.1	**Contacts and support service**			
Control 25	IoT users should only choose IoT devices and IoT services that provide contact information for support service.	To ensure security in the use of IoT device and service.	IoT user	
7.1.3.2	**Initial settings of IoT device and service**			
Control 26	Initial settings of IoT device and service should be applied correctly.	To ensure secure initial settings of IoT devices and service.	IoT user	
7.1.3.3	**Deactivate unused devices**			
Control 27	IoT devices should be deactivated and credentials revoked when they are no longer in use.	To reduce the security risks caused by the IoT device that is no longer used.	IoT user	
7.1.3.4	**Secure disposal or re-use of IoT device**			
Control 28	Data and licensed software stored in IoT device should be removed or securely overwritten prior to disposal or re-use.	To ensure information protection in disposal or re-use of IoT devices.	IoT user	
7.2	**Privacy controls**			
7.2.2	**Privacy controls for IoT service developer and IoT service provider**			
7.2.2.1	**Prevention of privacy invasive events**			
Control 29	Privacy enhancing capabilities should be built in the IoT devices and IoT services.	To prevent privacy invasive events in the provision and use of IoT devices and IoT services.	IoT service developer/ IoT service provider	
7.2.2.2	**IoT privacy by default**			
Control 30-1	Stakeholders in an IoT system should ensure that without any IoT user interaction, the strictest privacy settings apply by default.	To protect PII without the need of user intervention.	IoT service developer/ IoT service provider	
Control	Stakeholders in an IoT system	To protect PII without the	IoT service	

30-2	should ensure that the strictest privacy settings are applied by default, without any intervention of IoT user.	need of user intervention.	developer/ IoT service provider	
7.2.2.3	**Collection and use of personal data**			
Control 31-1	The IoT user should be provided with a privacy notice which states personal data collected by the IoT device and IoT service and purpose of its use.	To ensure use of personal data.	IoT service provider	
Control 31-2	Consent of the IoT user to the privacy notice should be obtained before collecting the personal data or changing the purpose of use.	To ensure consented collection and use of personal data.	IoT service provider	
7.2.2.4	**Verification of IoT functionality**			
Control 32	Independent verification of IoT device, data components and IoT service components should be supplied to provide visibility and assurance to all stakeholders that the IoT device or service is operating as per stated objectives.	To ensure WYSIWYG (What You Sees Is What You Get) of functionalities for IoT devices and services.	IoT service developer/ IoT service provider	
7.2.2.5	**Consideration of IoT users**			
Control 33	End users' privacy requirements and concerns should be addressed in designing the IoT device and service.	To ensure IoT users' privacy requirements and concerns are addressed in the IoT device and service and to build IoT users' trust.	IoT service developer/ IoT service provider	
7.2.2.6	**Management of IoT privacy controls**			
Control 34	The effectiveness of privacy controls in the IoT device and service should be reviewed, and new privacy risks be identified on a continuous basis considering the evolving privacy needs of end users and regulatory requirements.	To justify the effectiveness of privacy controls in IoT devices and services.	IoT service provider	
7.2.2.7	**Unique device identity**			
Control	IoT system developers (especially	To enable identification of the	IoT service	

35-1	device developer) should use a method to allow a unique identification of each IoT device.	IoT device suspected to be relevant to a cyber incident.	provider	
Control 35-2	IoT service providers should use, if required, a method to allow a unique mapping between a given IoT device and an IoT user.	To uniquely identify a mapping between IoT device and IoT user(s).	IoT service provider	
7.2.2.8	**Fail-safe authentication**			
Control 36	The system should ensure that implemented authentication cannot be bypassed, tampered, or falsified in any reasonable method.	Since the device (thing) is often not with the user and consequences of a wrong user connecting to device can cause serious harm in terms of safety, financial loss, health hazard etc. In case of traditional authentication service, the result of access is evident to the user since user is able to witness consequence of his/her action.	IoT service developer / IoT service provider	
7.2.2.9	**Minimization of indirect data collection**			
Control 37	Collection of data from indirect sources should be minimized or not collected at all.	To prevent data collection without the IoT users' participation and consent.	IoT service provider	
7.2.2.10	**Communication of privacy preferences**			
Control 38	User preferences of privacy controls should be only added, modified, or deleted when the authorized user is authenticated to the system.	Unlike in conventional scenarios whereby privacy preferences are known to the organization that collects PII, in case of IoT the same is not possible since there are multiple devices and services that need to access data.	IoT service provider	
7.2.2.11	**Verification of automated decision**			
Control 39	Automated decision provided by IoT services should be verified.	To avoid irreversible harm caused by erroneous automated decision made by an IoT device or system.	IoT service provider	
7.2.2.12	**Accountability for stakeholders**			
Control 40	Accountability for various stakeholders should be	To define responsibilities among stakeholders of IoT	IoT service developer /	

	established.	system. In the event of a data breach or data subject requests, which entity will respond, who will cater to data disclosure requests etc.	IoT service provider	
7.2.2.13	**Unlink ability of PII**			
Control 41	The IoT system should ensure that the PII of the user owning a device cannot be identified.	Prevent the collection of PII by monitoring an IoT device.	IoT service developer / IoT service provider	
7.2.2.14	**PII protection in IoT devices**			
Control 42	PII protection measures related to privacy risk in IoT devices should be appropriately managed and only disclosed to the parties that require them.	-	IoT service developer	
7.2.3	**Privacy controls for IoT user**			
7.2.3.1	**User Content**			
Control 43	Consent for use of personal data for the IoT device and service should be provided only after considering the necessity and its probable impact if there is a data breach. Consent should be withdrawn if the IoT output is no longer needed or if there is a concern with the IoT device or service.	To prevent use of PII by the IoT device and service without user's consent.	IoT user	
7.2.3.2	**Connecting with other devices and services**			
Control 44	Connection of IoT device and service with other devices or services should be allowed only if there is a valid need.	To ensure purposeful connection between IoT devices and services.	IoT user	
7.2.3.3	**Certification/validation of PII protection**			
Control 45	Certification or validation of privacy protection features with respect to the IoT device and service should be provided.	To ensure that users' PII will not be compromised when they opt for a certified/validated IoT device/ service.	IoT service developer / IoT service provider	