IoT Security Checklist

User Manual

JPCERT Coordination Center
June 27, 2019

Contents

1. Introduction

The Internet of Things (IoT) is a concept that refers to a distributed system consisting of a network of monitoring applications that collect information about the state of physical objects, and control applications that alter the state of objects based on the collected information and other elements. One application of this concept may involve newly providing communication capabilities to individual devices that measure temperatures of and control manufacturing equipment, and building a system that connects them to a network and leverages cloud services and so on, to make it possible to efficiently collect and aggregate production information within a vast factory. Devices newly equipped with communication capabilities as in this example are often called “IoT devices.” Recently, systems built with IoT devices based on the concept of IoT are gaining a lot of attention and are expected to spread rapidly in the years to come.

For example, according to the Ministry of Internal Affairs and Communications[[1]], the number of IoT devices will continue to trend upward worldwide, given the huge number of smartphones, communication equipment, and other similar devices in use as of 2017, and the rapid growth predicted in various areas, including: automobiles and transportation equipment, which are expected to see increasing adoption of IoT through the popularization of connected cars; healthcare, which is witnessing an expansion of the digital healthcare market; and industrial applications (factories, infrastructure, and logistics), where the number of smart factories and smart cities is growing.

Meanwhile, IoT devices as well as systems built based on the concept of IoT are constantly connected to a network, and in many cases, numerous IoT devices of the same type are connected to a network, which often makes it difficult to ensure security control of individual IoT devices.

Broadband routers for general consumers face the same situation as IoT devices in that countless devices are constantly connected to a network, making it difficult to ensure security control. In fact, a large number of security incidents involving broadband routers infected with the Mirai malware have been reported since around 2016. Improperly managed devices such as those that have not been updated with security patches addressing vulnerabilities, and those that are not configured with rigorous authentication settings for the administration console are susceptible to infection by the Mirai malware. Similar malware targeting IoT devices could emerge once these devices become widespread, giving rise to concerns about similar damage resulting from infection.

To ensure secure operation of IoT and other devices so that IoT can be used safely, these devices themselves must be secured against attacks by rooting out vulnerabilities and other measures.

[1] Ministry of Internal Affairs and Communications: Part 1 Special Feature: ICT as a Driver of Sustainable Growth in an Age of Declining Population
(http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd111200.html)

However, developers of IoT devices often put too much focus on creating new functions and forget to work on security design, as in the case of broadband routers infected with Mirai. This situation can be greatly improved by providing a security checklist that can be readily used to check security functions of IoT devices and having developers use them when designing devices.
Also, when users build a system using IoT devices, it is important to select constituent products that provide necessary security features. If users select an inappropriate product, it could be subject to cyber attacks, preventing the system from operating as expected or enabling the system to be used as a springboard for cyber attacks against third parties. Yet users also tend to focus on implementing the main functions, allowing the project to proceed without sufficient consideration of security features. This situation can also be avoided by providing a checklist that can be readily used to check security functions of IoT devices and encouraging users to use them when building systems.

To address these issues, JPCERT/CC is publishing an IoT security checklist (hereafter, “checklist”) developed in cooperation with the IoT security working group of the Japan Network Security Association (JNSA) and the University of Nagasaki. The checklist was developed to be simple and practical so that IoT device developers and users can use them easily. Check items were narrowed down to include only the basic, essential items and organized into a list describing what to check and why.

In this checklist, matters to be checked by developers and users are listed side by side for each check item. We chose this format to enable IoT device developers to consider users’ concerns, and users to note the developers’ considerations, so that their thoughts will converge to help realize safe use of IoT.

Since we made the checklist as simple as possible, it might not be clear how to use it at first. We therefore prepared this user manual to provide the necessary explanation. Chapter 2 gives an overview of the checklist, Chapter 3 explains how to use the checklist while providing examples, and Chapter 4 sheds light on the relationship between the checklist and other guides. Please read this user manual to help you understand how to use the checklist.

2. Overview of the Checklist and When to Use It

2.1. Overview of the Checklist

2.1.1. Components

The checklist lists 39 essential security functions that enable an entire system built based on the concept of IoT (hereafter, "IoT system") to be operated safely even in an environment where threats exist, along with background information on why they are necessary. The list is designed to be used when developing and deploying products that make up an IoT system.

By using this checklist to evaluate an IoT system that is under development or planned to be deployed, it is possible to determine whether the functions necessary to ensure security of the IoT system are provided, and identify any matters that need further consideration.

The checklist consists of the following components.

(1) IoT Security Checklist
A list of 39 check items to ensure a device is equipped with the security functions needed for an IoT system

(2) IoT Security Checklist Illustration Diagrams
A collection of diagrams to help illustrate the check items on the IoT Security Checklist

(3) IoT Security Checklist User Manual (this document)
A user manual on how to use the IoT Security Checklist

2.1.2. Overview of how to use the checklist

Use the checklist to evaluate security functions in the following four steps.

Step 1: Determine the primitives contained in the IoT device to be evaluated

Determine which primitives the target IoT device is equipped with. Note that a device may have multiple primitives. See 2.1.3 for details about primitives.

Step 2: Determine the items to be evaluated

The 39 security functions required in an IoT device can be grouped into eight categories—"user management," "software management," "security management," "access control," "unauthorized connection," "encryption," "system settings," and "notification"—but the checklist can be used to evaluate only some of these categories. In this step, select the categories you wish to evaluate. If you wish to evaluate all the items, select all categories.

Step 3: Extract the items that need to be checked

Leave only the items related to the primitives determined in Step 1 and categories selected in Step 2, and remove all other items from the checklist.

Step 4: Evaluate the security functions

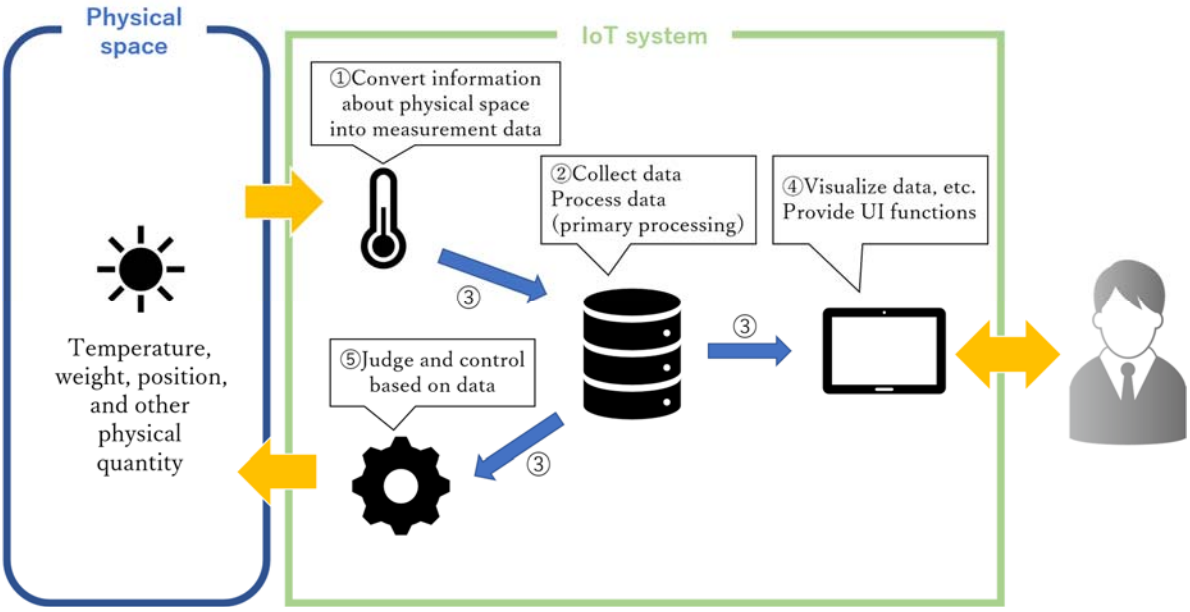
Conduct evaluation according to the checklist items extracted in Step 3, and referencing the IoT Security Checklist Illustration Diagrams.

We will explain how to use the checklist in greater detail in Chapter 3.

2.1.3. IoT system structure and primitives

In this section, we will discuss the primitives determined in Step 1 when using the checklist.

The checklist breaks down IoT systems into various elements according to functions and roles for the purpose of generic observation. These constituent elements are referred to as primitives. An IoT system can generally be broken down into the structure shown in [Figure 1], given that it is designed to collect information in physical space and control objects to change their state based on the collected information.


Physical space
IoT system
①Convert information about physical space into measurement data
②Collect data Process data (primary processing)
④Visualize data, etc. Provide UI functions
⑤Judge and control based on data
③
③
③
Temperature, weight, position, and other physical quantity

[Figure 1: How IoT systems operate]

① measures the state of physical space and converts it into measurement data, which is collected, processed (primary processing), and stored by ②. Users can access and configure data stored in ②, but the interface used for that purpose is provided by ④, ⑤ initiates action to change the state of objects based on the data stored in ②. As these functions are distributed over the network, a communication function (③) is also needed to connect them.

Following the example of NIST SP800-183, the checklist refers to ① as sensors, ② as aggregators, ③ as communication channels, ④ as eUtilities, ⑤ as decision triggers, and the building blocks (① through ⑤) that make up an IoT system as primitives.

Sensor	A function or device for measuring temperatures, acceleration, weight, sound, positions, etc.
Aggregator	A function or device for aggregating data captured by sensors
Communication Channel	A communication path or network for sending and receiving data
eUtility	An interface for viewing data and configuring settings
Decision Trigger	A function for calculating data and triggering an action based on the result

Note that Figure 1 is a simplified diagram, and that actual IoT systems may contain multiple sensors, have aggregators and eUtilities distributed over edge and cloud environments, and take various other forms. Decision triggers do not exist in IoT systems that only collect data and do not have control functions.

IoT devices to be evaluated with this checklist should have at least one of these primitives. The security functions required vary depending on which primitives are implemented on the IoT device, so the primitives must be identified correctly.

2.2. When to Use the Checklist

2.2.1. When to be used by product developers and expected benefits

When:

(A) Use the checklist to verify whether the necessary security functions are incorporated when considering the basic functions during the planning stage or early design stage of an IoT system or its constituent product.

Expected benefits:

- By using the checklist in early stages of product development, it is possible to systematically check whether the essential security functions are provided and correct any problems early.
- A standard can be created for the security functions to be provided to a product, and this standard can be applied to other products developed by the company to ensure a certain level of security.

- Problems can be identified at an early stage, reducing the amount of rework in the development process, as well as the risk of releasing a product without all the necessary security functions and making the user susceptible to cyber attacks.

2.2.2. When to be used by product users and expected benefits

When:

(A) When building an IoT system by combining existing products, once candidate products have been narrowed down, use the checklist to see whether they are fully equipped with the necessary security functions.

(B) Use the checklist to confirm that each constituent product of an existing IoT system is running the necessary security functions. If any function is found to be missing, consider countermeasures such as adding new equipment.

Expected benefits:

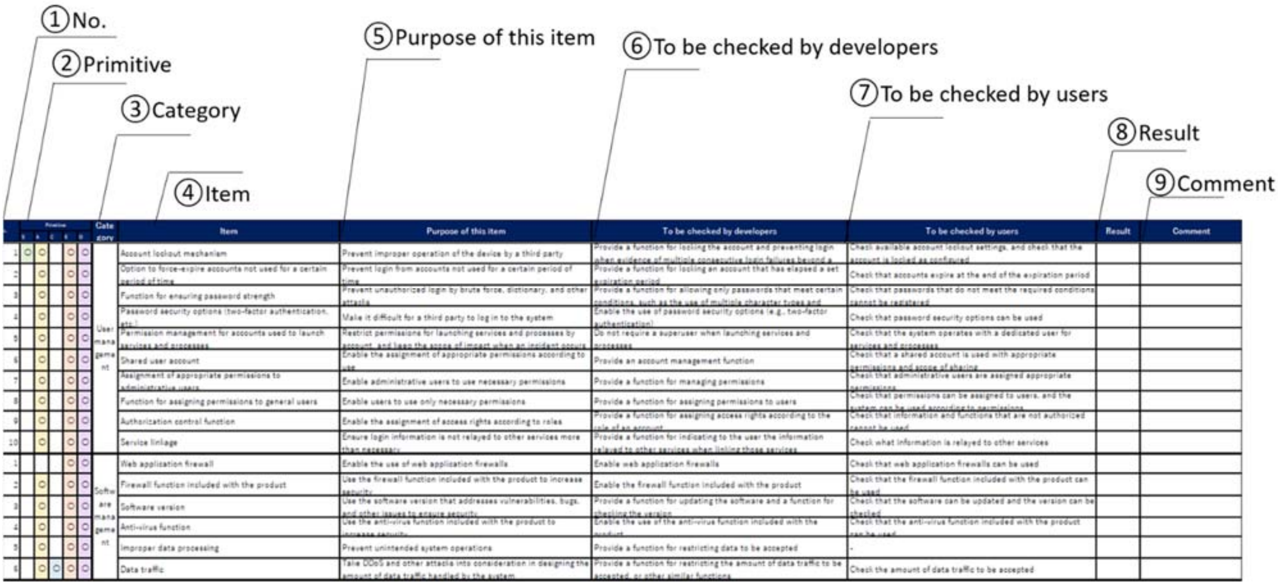
- (A): By selecting products that are fully equipped with security functions, the risk of cyber attacks can be reduced.
- (B): By identifying functions that an existing IoT system is lacking and making up for the gap with additional measures, the system can be operated with all the necessary security functions, reducing the risk of cyber attacks.

3. How to Use the Checklist

3.1. How Product Developers Should Use the Checklist

3.1.1. Structure of the checklist

The checklist is organized into a table as shown in [Figure 2]. Each row of the table corresponds to individual check items, and consists of nine columns as shown in [Table 1].



No.	Primitive					Category	Item	Purpose of this item	To be checked by developers	To be checked by users	Result	Comment
	S	A	C	E	R							
1	○	○		○	○	User management	Account lockout mechanism	Prevent improper operation of the device by a third party	Provide a function for locking the account and preventing login when evidence of multiple consecutive login failures beyond a	Check available account lockout settings, and check that the account is locked as configured		
2		○		○	○		Option to force-expire accounts not used for a certain period of time	Prevent login from accounts not used for a certain period of time	Provide a function for locking an account that has elapsed a set expiration period	Check that accounts expire at the end of the expiration period		
3		○		○	○		Function for ensuring password strength	Prevent unauthorized login by brute force, dictionary, and other attacks	Provide a function for allowing only passwords that meet certain conditions, such as the use of multiple character types and	Check that passwords that do not meet the required conditions cannot be registered		
4		○		○	○		Password security options (two-factor authentication, etc.)	Make it difficult for a third party to log in to the system	Enable the use of password security options (e.g., two-factor authentication)	Check that password security options can be used		
5		○		○	○		Permission management for accounts used to launch services and processes	Restrict permissions for launching services and processes by account, and keep the scope of impact when an incident occurs	Do not require a superuser when launching services and processes	Check that the system operates with a dedicated user for services and processes		
6		○		○	○		Shared user account	Enable the assignment of appropriate permissions according to use	Provide an account management function	Check that a shared account is used with appropriate permissions and scope of sharing		
7		○		○	○		Assignment of appropriate permissions to administrative users	Enable administrative users to use necessary permissions	Provide a function for managing permissions	Check that administrative users are assigned appropriate permissions		
8		○		○	○		Function for assigning permissions to general users	Enable users to use only necessary permissions	Provide a function for assigning permissions to users	Check that permissions can be assigned to users, and the system can be used according to permissions		
9		○		○	○		Authorization control function	Enable the assignment of access rights according to roles	Provide a function for assigning access rights according to the role of an account	Check that information and functions that are not authorized cannot be used		
10		○		○	○		Service linkage	Ensure login information is not relayed to other services more than necessary	Provide a function for indicating to the user the information relayed to other services when linking those services	Check what information is relayed to other services		
1				○	○	Software management	Web application firewall	Enable the use of web application firewalls	Enable web application firewalls	Check that web application firewalls can be used		
2		○		○	○		Firewall function included with the product	Use the firewall function included with the product to increase security	Enable the firewall function included with the product	Check that the firewall function included with the product can be used		
3		○		○	○		Software version	Use the software version that addresses vulnerabilities, bugs, and other issues to ensure security	Provide a function for updating the software and a function for checking the version	Check that the software can be updated and the version can be checked		
4		○		○	○		Anti-virus function	Use the anti-virus function included with the product to increase security	Enable the use of the anti-virus function included with the product	Check that the anti-virus function included with the product can be used		
5		○		○	○		Improper data processing	Prevent unintended system operations	Provide a function for restricting data to be accepted	-		
6		○	○	○	○		Data traffic	Take DDoS and other attacks into consideration in designing the amount of data traffic handled by the system	Provide a function for restricting the amount of data traffic to be accepted, or other similar functions	Check the amount of data traffic to be accepted		

[Figure 2 IoT Security Checklist]

[Table 1 Elements of the IoT Security Checklist]

No.	Item Name	Description
①	No.	Number of the check item
②	Primitive	The IoT primitives explained in “2.1.3. IoT system structure and primitives” correspond to each initial. (S: Sensor, A: Aggregator, E: eUtility, D: Decision Trigger, C: Communication Channel) The items that need to be checked for each primitive are marked with a circle “○”.
③	Category	39 check items are grouped into one of the following 8 categories: (1) User management (2) Software management (3) Security management (4) Access control (5) Unauthorized connection (6) Encryption (7) System settings (8) Notification
④	Item	
⑤	Purpose of this item	Security requirements corresponding to each check item
⑥	To be checked by developers	Matters to be checked by product developers. Write the check results in the Result column.
⑦	To be checked by users	Matters to be checked by product users. Write the check results in the Results column.
⑧	Result	Column for writing in the check results
⑨	Comment	If the check result is not “Pass” but you decide that there is no problem, write the reason in this column.

3.1.2. How to Use the Checklist

Step 1

Determine which primitive(s) the product to be developed corresponds to within the entire IoT system. See “2.1.3. IoT system structure and primitives” for details.

For example, a watch-type wearable device that is capable of measuring heart beats and steps corresponds to “Sensor,” but it also corresponds to “Aggregator” because it aggregates heart rate and step data and communicates with an external server.

Step 2

If you wish to check certain types of security functions from among the eight categories (user management, software management, security management, access control, unauthorized connection, encryption, system settings, and notification), select the relevant items to be checked.

If you do not wish to select specific categories, check all items.

For example, if you wish to check for necessary functions for the access control of a network camera:

→ Check items No. IV-1 to No. IV-4 under the access control category.

If you wish to check all items:

→ Check No. I-1 to No. VIII-2.

Step 3

Extract the items that need to be checked.

- Select only the check items with a circle “〇” for the primitive determined in Step 1. Ignore items without a circle “〇”.
- The items that correspond to the primitive(s) determined within the categories selected in Step 2 are the security functions that need to be checked.

Step 4

Check whether the function described within the “To be checked by developers” column is or will be implemented, and write “Pass” or “Fail” in the Result column. If you have trouble understanding the description in the “To be checked by developers” column, please refer to the IoT Security Checklist Illustration Diagrams. If the check result was “Fail” for any item, consider matters such as whether that function can be implemented or substituted with a different function, and write the findings in the “Comment” column. This comment will serve as a reference when preparing an explanatory document for other parties or a manual.

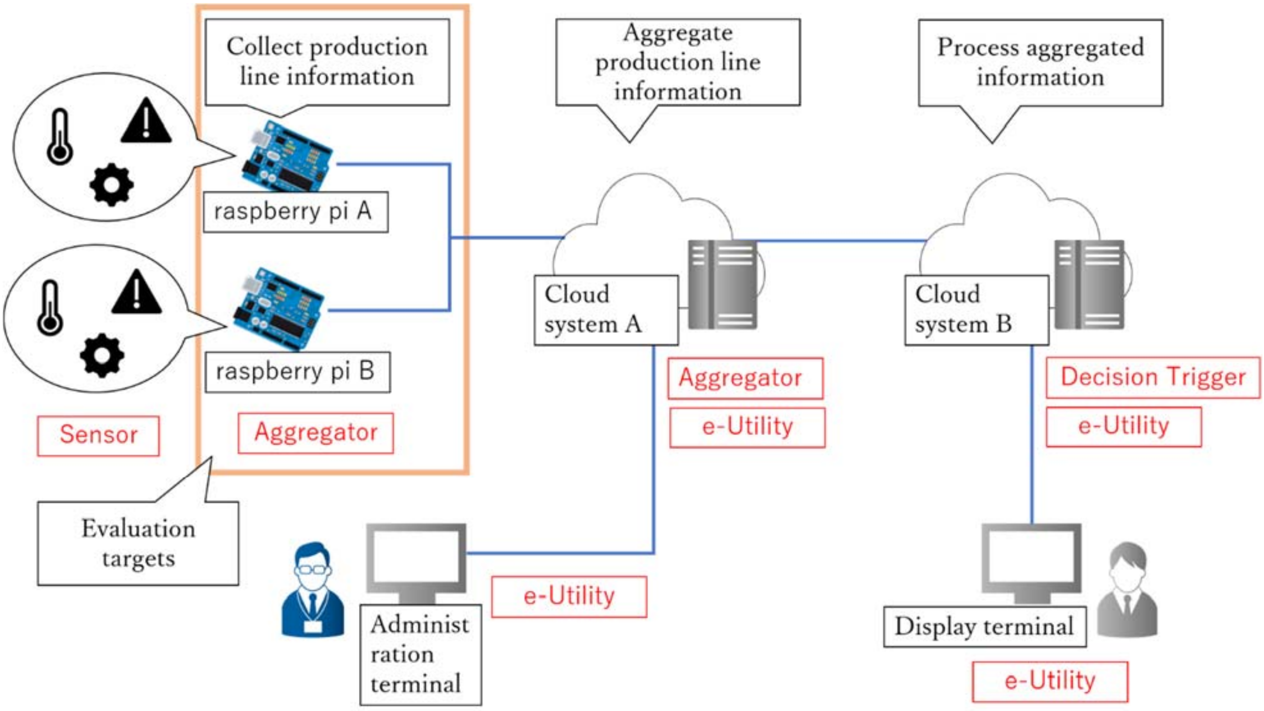
3.1.3. Application example

As a case example for using the checklist, we will discuss a case where the checklist is used to review the security of a production information collection system in a factory.

3.1.3.1. About the products to be evaluated with the IoT Security Checklist

This system uses Raspberry Pi to collect production information for each production line involved in the manufacturing of product P, aggregates the production information on cloud system A, then processes the information on cloud system B to make it accessible on data display devices.

[Figure 3] is a structure chart that breaks down the system into element levels, with the type of each element indicated referencing “2.1.3. IoT system structure and primitives.”



[Figure 3 Production information collection system in a factory]

In the case of this production information collection system, both cloud systems have a screen display function for users, so it is assumed that they serve the roles of multiple primitives (i.e., cloud system A, aggregator and eUtility, and cloud system B, decision trigger and eUtility).

3.1.3.2. Applying the checklist to a factory production information collection system

In this case example, the system’s Raspberry Pi A and B were evaluated using the checklist.
Based on the specifications, the items indicated in [Table 2] were deemed to be “Fail.”

[Table 2 Applicable items on the checklist]

Category	Item
User management	Password security options (two-factor authentication, etc.)
	Permission management for accounts used to launch services and processes
	Service coordination
Software management	Anti-virus function
	Improper data processing
	Data traffic
Security management	Session management (Cookie settings)
	Session management (URL rewriting)
	Session management (Issuance of session ID when logging in and processing important confirmation)
Access control	Default ports for remote access
Unauthorized connection	UPnP
Encryption	Data encryption function
	Encryption method

Of the above items, it was decided not to implement the following functions since the information handled by the system is not regarded as strictly confidential in the company rules.

- User management: Password security options (two-factor authentication, etc.)
- Encryption: Data encryption function

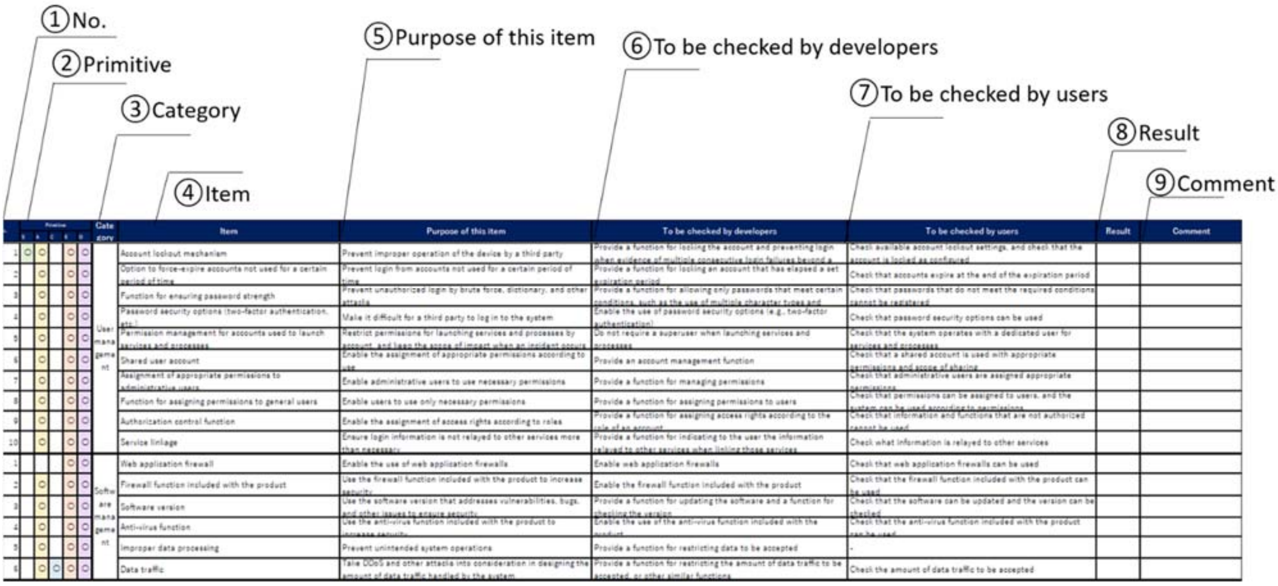
Of the remaining items, it was deemed that the following can actually be implemented and need to be considered for adding.

- User management: Service coordination
- Software management: Improper data processing
- Software management: Data traffic
- Security management: Session management (Cookie settings)
- Access control: Default ports for remote access
- Encryption: Encryption method

3.2. How Product Users Should Use the Checklist

3.2.1. Name of each item

The checklist is organized into a table as shown in [Figure 4]. Each row of the table corresponds to individual check items, and consists of nine columns as shown in [Table 3].



No.	Primitive					Category	Item	Purpose of this item	To be checked by developers	To be checked by users	Result	Comment
	[illegible]	A	C	E	R							
1	○	○		○	○	User management	Account lockout mechanism	Prevent improper operation of the device by a third party	Provide a function for locking the account and preventing login when evidence of multiple consecutive login failures beyond a	Check available account lockout settings, and check that the account is locked as configured		
2		○		○	○		Option to force-expire accounts not used for a certain period of time	Prevent login from accounts not used for a certain period of time	Provide a function for locking an account that has elapsed a set expiration period	Check that accounts expire at the end of the expiration period		
3		○		○	○		Function for ensuring password strength	Prevent unauthorized login by brute force, dictionary, and other attacks	Provide a function for allowing only passwords that meet certain conditions, such as the use of multiple character types and	Check that passwords that do not meet the required conditions cannot be registered		
4		○		○	○		Password security options (two-factor authentication, etc.)	Make it difficult for a third party to log in to the system	Enable the use of password security options (e.g., two-factor authentication)	Check that password security options can be used		
5		○		○	○		Permission management for accounts used to launch services and processes	Restrict permissions for launching services and processes by account, and keep the scope of impact when an incident occurs	Do not require a superuser when launching services and processes	Check that the system operates with a dedicated user for services and processes		
6		○		○	○		Shared user account	Enable the assignment of appropriate permissions according to use	Provide an account management function	Check that a shared account is used with appropriate permissions and scope of sharing		
7		○		○	○		Assignment of appropriate permissions to administrative users	Enable administrative users to use necessary permissions	Provide a function for managing permissions	Check that administrative users are assigned appropriate permissions		
8		○		○	○		Function for assigning permissions to general users	Enable users to use only necessary permissions	Provide a function for assigning permissions to users	Check that permissions can be assigned to users, and the system can be used according to permissions		
9		○		○	○		Authorization control function	Enable the assignment of access rights according to roles	Provide a function for assigning access rights according to the role of an account	Check that information and functions that are not authorized cannot be used		
10		○		○	○		Service linkage	Ensure login information is not relayed to other services more than necessary	Provide a function for indicating to the user the information relayed to other services when linking those services	Check what information is relayed to other services		
1				○	○	Software management	Web application firewall	Enable the use of web application firewalls	Enable web application firewalls	Check that web application firewalls can be used		
2		○		○	○		Firewall function included with the product	Use the firewall function included with the product to increase security	Enable the firewall function included with the product	Check that the firewall function included with the product can be used		
3		○		○	○		Software version	Use the software version that addresses vulnerabilities, bugs, and other issues to ensure security	Provide a function for updating the software and a function for checking the version	Check that the software can be updated and the version can be checked		
4		○		○	○		Anti-virus function	Use the anti-virus function included with the product to increase security	Enable the use of the anti-virus function included with the product	Check that the anti-virus function included with the product can be used		
5		○		○	○		Improper data processing	Prevent unintended system operations	Provide a function for restricting data to be accepted	-		
6		○	○	○	○		Data traffic	Take DDoS and other attacks into consideration in designing the amount of data traffic handled by the system	Provide a function for restricting the amount of data traffic to be accepted, or other similar functions	Check the amount of data traffic to be accepted		

[Figure 4 IoT Security Checklist]

[Table 3 Elements of the IoT Security Checklist]

No.	Item Name	Description
①	No.	Number of the check item
②	Primitive	The IoT system's primitives explained in "2.1.3. IoT system structure and primitives" correspond to each initial. (S: Sensor, A: Aggregator, E: eUtility, D: Decision Trigger, C: Communication Channel) The items that need to be checked for each primitive are marked with a circle "○".
③	Category	39 check items are grouped into one of the following 8 categories: (1) User management (2) Software management (3) Security management (4) Access control (5) Unauthorized connection (6) Encryption (7) System settings (8) Notification
④	Item	
⑤	Purpose of this item	Security requirements corresponding to each check item
⑥	To be checked by developers	Matters to be checked by product developers. Write the check results in the Result column.
⑦	To be checked by users	Matters to be checked by product users. Write the check results in the Results column.
⑧	Result	Column for writing in the check results
⑨	Comment	If the check result is not "Pass" but you decide that there is no problem, write the reason in this column.

3.2.2. How to Use the Checklist

Step 1

Determine which primitive(s) the product to be used corresponds to within the entire IoT system.
See "2.1.3. IoT system structure and primitives" for details.

For example, a watch-type wearable device that is capable of measuring heart beats and steps corresponds to "Sensor," but it also corresponds to "Aggregator" because it aggregates heart rate and step data and communicates with an external server.

Step 2

If you wish to check certain types of security functions from among the eight categories (user management, software management, security management, access control, unauthorized connection, encryption, system settings, and notification), select the relevant items to be checked.
If you do not wish to select specific categories, check all items.

For example, if you wish to check for necessary functions for the access control of a network camera:

→ Check items No. IV-1 to No. IV-4 under the access control category.

If you wish to check all items:

→ Check No. I-1 to No. VIII-2.

Step 3

Extract the items that need to be checked.

- Select only the check items with a circle "○" for the primitive determined in Step 1.

Ignore items without a circle "○".

- The items that correspond to the primitive(s) determined within the categories selected in Step 2 are the security functions that need to be checked.

Step 4

Check whether the product to be evaluated fulfills the descriptions stated in "To be checked by users."

- Check whether the product is equipped with the functions described in "To be checked by users."
- Check the items referencing the IoT Security Checklist Illustration Diagrams attached to the IoT Security Checklist.

Write a check mark in the "Result" column for items that meet the description, and write the reason in the "Comment" column for those that do not.

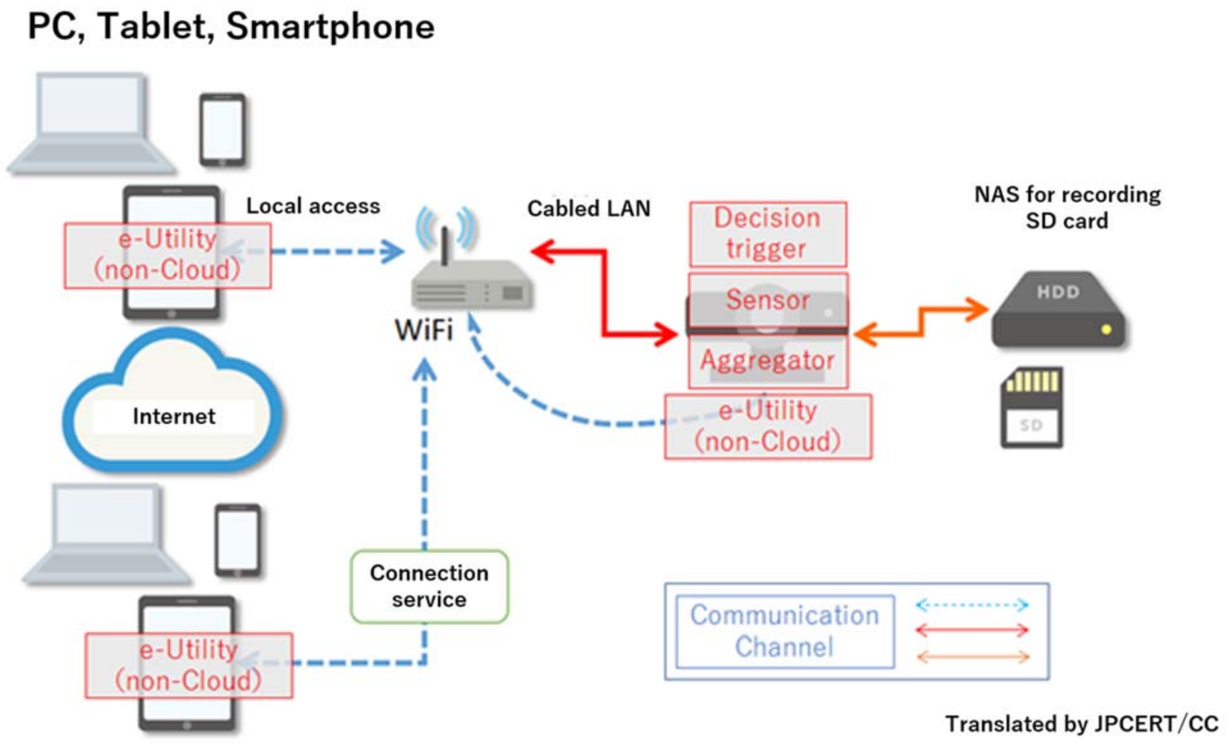
- For items without a check mark, consider matters such as whether the function can be implemented at the time of development or complemented with another function, and write the findings in the "Comment" column to be used as reference when preparing an explanatory document or a manual.

3.2.3. Application example

As a case example for using the checklist, we will discuss a case where a user who is considering to introduce a commercially available network camera for the purpose of remotely monitoring his home for security reasons conducted an evaluation when considering the security measures needed in the product.

3.2.3.1. About the products to be evaluated with the IoT Security Checklist

The following diagram shows the configuration of each function of the network camera categorized in the evaluation.
[Figure 5] was created by referencing the web camera example in the "IoT Security Guide for Consumers" issued by JNSA.



[Figure 5 Categorization by primitive in the case of a network camera]

Reference: "IoT Security Guide for Consumers," Japan Network Security Association (JNSA)

In the configuration of the network camera in this example, the camera itself has many functions and therefore can have multiple primitives assigned.

- Web camera (Sensor, Aggregator, eUtility, Decision Trigger)
- Smartphones, PCs (eUtility)

- Wi-Fi, internal bus (Communication Channel)

3.2.3.2. Applying the checklist to a network camera

In this example, the checklist was used to evaluate the eUtility portion of the network camera.
[Table 4] lists the functions that the network camera the user is considering to introduce was found to be missing after checking the items on the checklist against the product's user manual.

[Table 4 Applicable items on the checklist]

Category	Item
User management	Option to force-expire passwords past the expiration date
	Function for ensuring password strength
	Password security options (two-factor authentication, etc.)
	Permission management for accounts used to launch services and processes
Software management	Firewall function included with the product
	Anti-virus function
Access control	Access by uncontrolled physical means
	Default ports for remote access
	Wireless communication security (WPS)
Unauthorized connection	Restriction of network ports
	UPnP

In light of the above results, three items were chosen for discussion in introducing the network camera that was evaluated.

(1) Enhancement of authentication function

Network cameras' authentication function is often not configured rigorously enough with many products. For example, there are products shipped with the same simple authentication password set for all devices, and with some of these products the password is made available on the Internet. While developers must consider a way to prevent the devices from being configured with the same password, they must also devise a specification that only allows a strong password to be set. It is also preferable to consider enhancing authentication by implementing two-factor authentication and the like. Users should also choose products that allow a strong password to be set or those that offer options such as two-factor authentication, and configure the products with appropriate settings.

(2) Restriction of external access

Network cameras are sometimes accessible to external parties without the user's knowledge. Attackers may target such devices' default ports for remote access to cause malware infection, so users must consider external access control or changing the ports for remote access. Users should also be aware that UPnP (Universal Plug and Play) and other similar functions may be used to gain access to a device in a NAT environment from the Internet. Developers must consider functions for controlling external access and changing ports for remote access, as well as specifications that allow disabling UPnP.

(3) Using available product functions

If available, use of firewall and anti-virus functions should also be considered as a countermeasure. Depending on the product's specifications, it may be difficult to implement these functions as eUtilities. In such cases, consider implementing them elsewhere. Attackers tend to persistently attack known security holes, so developers must implement a function for updating the firmware to fix vulnerabilities and so on when they are published, and users must keep their products up-to-date.

4. Relationship between the Checklist and Other Guides

The items of the checklist were created based on IT security evaluation documents in addition to security evaluation documents for IoT. This is because some IoT systems may be configured using existing IT technologies—for example, an interface implemented for eUtility and so on featured in a product may use a communication protocol similar to the web, such as access via a web application or API—and we thought security evaluation designed for IT will be helpful.

In creating the checklist, we chose the following documents as our main reference, which offer particularly detailed explanations on how to conduct security evaluations, out of the many IoT/IT security evaluation documents available.

IoT Security Guidance
OWASP
https://www.owasp.org/index.php/IoT_Security_Guidance

The Penetration Testing Execution Standard
Penetration Testing Execution Standard Group
http://www.pentest-standard.org/index.php/Main_Page

We identified common items in these two documents and used them as standard items in our discussions. These documents summarize basic items in thinking about IT and IoT security, and many of them are covered in both documents. For these reasons, we believe these common items attract a lot of attention and should be given high priority.

Nevertheless, we do not suggest in this document that all of these items are requirements that must be fulfilled. In other words, we believe these high-profile items are recommended items that need to be discussed and considered for priority measures and so on.

This is because products and systems have other standards and requirements to meet and items are weighted accordingly. When applying the checklist, it should be considered together with those other requirements.

While the checklist summarizes basic items to be considered with regard to the security of IoT systems, some issues may warrant prioritizing a solution based on safety requirements rather than security requirements. In this document, we take the standpoint that it is appropriate to consider both safety and security. As such, we recommend considering both aspects simultaneously, rather than prioritizing either one, in ensuring appropriate measures are incorporated into final products and systems.

Acknowledgment

We are greatly indebted to Mr. Masato Matsuoka, leader of JNSA's IoT security working group, and Dr. Masahiko Kato, professor at the University of Nagasaki, for their invaluable help in reviewing this document and creating the checklist. We also received insightful advice from the members of JNSA's IoT security working group and many other collaborators in considering the points of discussion. We thank everyone who were involved in preparing this document.

References

・IoT Security Guidance
OWASP
https://www.owasp.org/index.php/IoT_Security_Guidance

・The Penetration Testing Execution Standard
Penetration Testing Execution Standard Group
http://www.pentest-standard.org/index.php/Main_Page

・NIST Special Publication 800-183 Networks of ‘Things
NIST
https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-183.pdf

・NIST SP 800-160 Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems
NIST
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160.pdf

・IoT Security Guide for Consumers
Japan Network Security Association (JNSA)
https://www.jnsa.org/result/iot/

・Overview of the Internet of things
International Telecommunication Union (ITU)
https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060

・IoT Security Guidelines (Ver 1.0)
Ministry of Economy, Trade and Industry/Ministry of Internal Affairs and Communications
http://www.iotac.jp/wp-content/uploads/2016/01/IoT-Security-Guidelines_ver.1.0.pdf

・IoT Safety/Security Development Guidelines (Second Edition)
Information-technology Promotion Agency, Japan (IPA)
https://www.ipa.go.jp/sec/reports/20160324.html

・The STRIDE Threat Model
Microsoft
https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

- The WASC Threat Classification v2.0
  WASC Projects
  http://projects.webappsec.org/w/page/13246978/Threat%20Classification

- OWASP Testing Guide v4 Table of Contents
  OWASP
  https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Cont

- If you would like to quote, reprint, or redistribute this document, please contact Public Relations (pr@jpcert.or.jp).
- JPCERT/CC assumes no responsibility for any loss or damage that may result from information contained in this document.
- Please note that fulfilling all items on this checklist does not guarantee conformity to any standard or international standard, or mean that IoT security measures are foolproof.