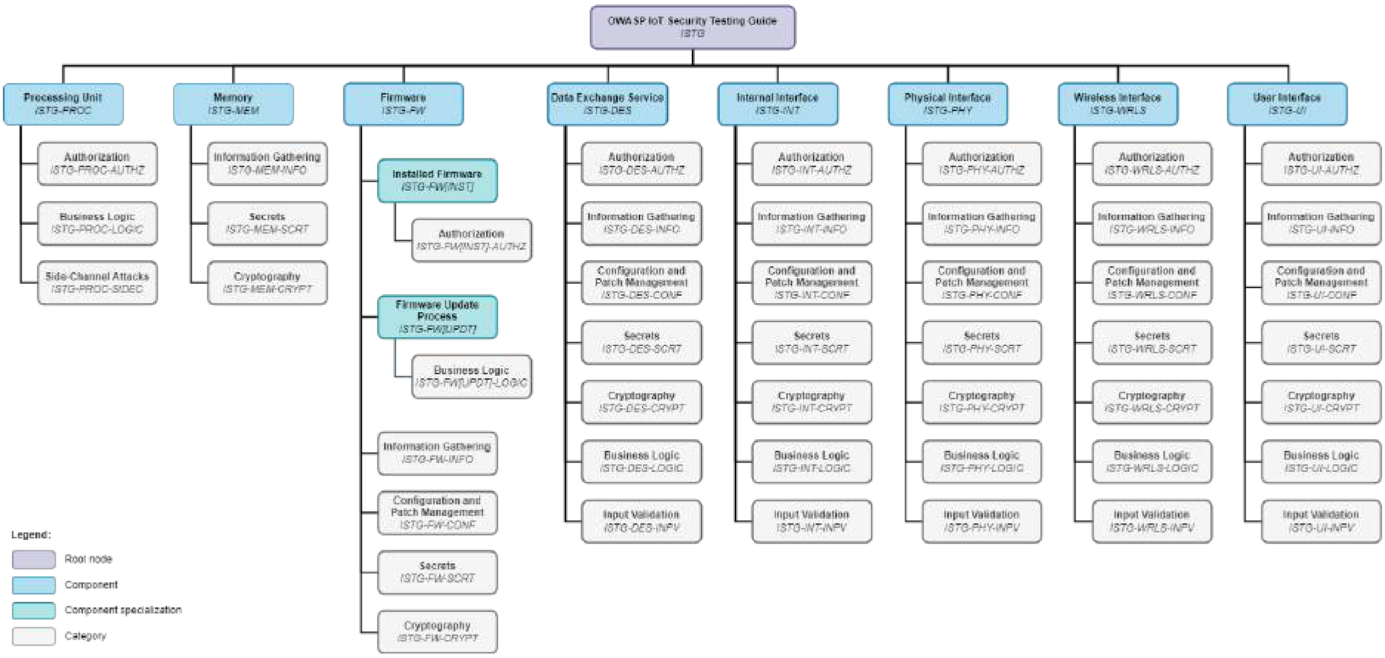
OWASP IoT Security Testing Guide

openssf best practices in progress 88%

The OWASP IoT Security Testing Guide provides a comprehensive methodology for penetration tests in the IoT field offering flexibility to adapt innovations and developments on the IoT market while still ensuring comparability of test results. The guide provides an understanding of communication between manufacturers and operators of IoT devices as well as penetration testing teams that's facilitated by establishing a common terminology.

Security assurance and test coverage can be demonstrated with the overview of IoT components and test case categories applicable to each below. The methodology, underlying models, and catalog of test cases present tools that can be used separately and in conjunction with each other.


OWASP IoT Security Testing Guide ISTG
Processing Unit ISTG-PROC
Authorization ISTG-PROC-AUTHZ
Business Logic ISTG-PROC-LOGIC
Side-Channel Attacks ISTG-PROC-SIDEC
Memory ISTG-MEM
Information Gathering ISTG-MEM-INFO
Secrets ISTG-MEM-SCRT
Cryptography ISTG-MEM-CRYPT
Firmware ISTG-FW
Installed Firmware ISTG-FW[INST]
Authorization ISTG-FW[INST]-AUTHZ
Firmware Update Process ISTG-FW[UPDT]
Business Logic ISTG-FW[UPDT]-LOGIC
Information Gathering ISTG-FW-INFO
Configuration and Patch Management ISTG-FW-CONF
Secrets ISTG-FW-SCRT
Cryptography ISTG-FW-CRYPT
Data Exchange Service ISTG-DES
Authorization ISTG-DES-AUTHZ
Information Gathering ISTG-DES-INFO
Configuration and Patch Management ISTG-DES-CONF
Secrets ISTG-DES-SCRT
Cryptography ISTG-DES-CRYPT
Business Logic ISTG-DES-LOGIC
Input Validation ISTG-DES-INPV
Internal Interface ISTG-INT
Authorization ISTG-INT-AUTHZ
Information Gathering ISTG-INT-INFO
Configuration and Patch Management ISTG-INT-CONF
Secrets ISTG-INT-SCRT
Cryptography ISTG-INT-CRYPT
Business Logic ISTG-INT-LOGIC
Input Validation ISTG-INT-INPV
Physical Interface ISTG-PHY
Authorization ISTG-PHY-AUTHZ
Information Gathering ISTG-PHY-INFO
Configuration and Patch Management ISTG-PHY-CONF
Secrets ISTG-PHY-SCRT
Cryptography ISTG-PHY-CRYPT
Business Logic ISTG-PHY-LOGIC
Input Validation ISTG-PHY-INPV
Wireless Interface ISTG-WRLS
Authorization ISTG-WRLS-AUTHZ
Information Gathering ISTG-WRLS-INFO
Configuration and Patch Management ISTG-WRLS-CONF
Secrets ISTG-WRLS-SCRT
Cryptography ISTG-WRLS-CRYPT
Business Logic ISTG-WRLS-LOGIC
Input Validation ISTG-WRLS-INPV
User Interface ISTG-UI
Authorization ISTG-UI-AUTHZ
Information Gathering ISTG-UI-INFO
Configuration and Patch Management ISTG-UI-CONF
Secrets ISTG-UI-SCRT
Cryptography ISTG-UI-CRYPT
Business Logic ISTG-UI-LOGIC
Input Validation ISTG-UI-INPV
Legend:
Root node
Component
Component specialization
Category

- 🔔 Click here to read the OWASP ISTG 📖 📚 🔔
- ✅ Get the latest ISTG Checklists ✅
- 📝 🔍 Contribute to ISTG

Table of Contents

Related Work

The concepts, models and test steps presented in the OWASP IoT Security Testing Guide are based on the master's thesis **"Development of a Methodology for Penetration Tests of Devices in the Field of the Internet of Things"** by Luca Pascal Rotsch.

Test cases were derived from the following public sources:

- OWASP **"Web Security Testing Guide"**
- OWASP **"Firmware Security Testing Methodology"**
- OWASP **"Mobile Security Testing Guide"**
- **"IoT Pentesting Guide"** by Aditya Gupta
- **"IoT Penetration Testing Cookbook"** by Aaron Guzman and Aditya Gupta
- **"The IoT Hacker's Handbook"** by Aditya Gupta
- **"Practical IoT Hacking"** by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- further sources are referenced in the respective test cases

We also like to thank our collaborators and supporters (see Project Collaborators and Acknowledgements)!

1. Introduction

Motivation

The networking of a multitude of different devices towards the Internet of Things (IoT) poses new challenges for manufacturers and operators of respective solutions. Due to the interconnection of many different technologies, standards and protocols, a considerable amount of effort is necessary to build up and maintain a homogeneous level of network security, data security and IT security in general. Additionally, since the IoT field is changing and developing quickly, manufacturers and operators must continuously monitor potential threats to their devices and networks.

While conventionally networked computer systems can be secured with established methods, e.g., restricting physical and authorization access to networks and important systems, these methods can be difficult to apply to IoT devices and ecosystems due to the above-mentioned heterogeneity and new network layouts. Compared to conventional computer networks, IoT infrastructures can be very wide-spread. Even though the back end infrastructure might be similar to conventional computer networks, IoT devices could be located at an arbitrary location, possibly even outside of a secure zone of the operator. In some cases, the devices are even physically accessible to third parties and potential attackers, e.g., connected cars, smart home devices or package stations. Hence, every IoT device represents a potential threat to user data and the entire infrastructure since a single manipulated device is sufficient to endanger the entire ecosystem.

In order to reduce the risk of successful attacks, manufacturers and operators should periodically assess the security level of their IoT solutions. An instrument for this purpose is penetration testing. The goal of a penetration test is to identify security vulnerabilities within IoT solutions. The results can be used to address the detected vulnerabilities and thus strengthen the security level.

Challenges

Within the context of penetration tests, it is important that the test procedure is transparent. Otherwise, the manufacturer or operator might not be able to understand the meaning of the test results to the full extent and could draw wrong conclusions. Furthermore, test results have to be reproducible, so that on the one hand the developers can replicate how a vulnerability was exploited in order to craft a sufficient fix and on the other hand to enable a proper retest once the fix has been applied.

Testing methodologies have been developed in order to make test procedures and results comparable and to ensure that the results of two or more testers, who perform the same test of the same target, do not differ. These well-known methodologies define a common approach for performing tests, including key aspects of testing and test cases, which have to be considered during a test. Unfortunately, only a few, not yet complete test methodologies for IoT penetration tests exist at the moment of writing this guide. Furthermore, these methodologies only focus on a specific technological area or are currently in an early development phase.

Goals

In order to solve the above-mentioned challenges, the aim of this guide is to develop a methodology for penetration tests of end devices in the IoT field, including general key aspects of testing.

The methodology should:

- be flexibly expandable so that more detailed test cases for certain technologies can be added later on (*expandability*).
- enable the comparison of test procedure (test steps/cases) and results regardless of specific technologies or device types (*comparability*).
- serve the purpose of a common language between manufacturers/operators and penetration testing service providers, meaning that it should facilitate the communication between both parties by establishing a comprehensible terminology (*comprehensibility*).
- be efficient, so that it can be used as a supporting instrument by penetration testing teams without requiring major changes to any established workflows or additions of any new steps or testing phases (*efficiency*).

Intended Audience

As the name suggests, the OWASP IoT Security Testing Guide is mainly intended to be used by penetration testers and security analysts in the IoT, hardware and embedded fields. However, others might benefit from the concepts and test cases introduced in this guide as well:

Builder

- **Manufacturers of IoT devices** (e.g., architects, engineers, developers and managers) can use the contents of this guide to get an understanding of potential issues and vulnerabilities that might affect their products. Since vulnerable products can lead to various kinds of damages for the manufacturer (financial loss, loss of reputation, etc.), there should be an interest in understanding how a certain product could be vulnerable in any given context or operational environment. By increasing the awareness and understanding early on in the design and development process, it is possible to improve product security in the long term while keeping the respective costs as low as possible.

Breaker

- **Penetration testers and bug bounty researchers** can use the concepts introduced in 2. IoT Security Testing Framework to plan their tests and define the test scope, test conditions and test approach. While performing the test, the test cases in 3. Test Case Catalog and the respective Checklists can be used:
  - a) as a guide that shows which aspects should be tested, why they should be tested, how they should be tested and how potential issues could be mitigated as well as
  - b) to keep track of the test completion status, making sure that all relevant aspects have been examined.
- **Security consultants and security managers** can use this guide and its contents as a common foundation for working with their teams and clients as well as communicating with any of the stakeholders mentioned above. Especially the terminology and structure defined in this guide should help to facilitate collaboration across different teams and organizations.

Defender

- **Operators of IoT devices** (e.g., users) can use this guide in a similar fashion as manufacturers. However, the operators who run IoT devices usually have no or very little influence on the design and development process. Hence, their focus is more directed towards understanding how a device might be vulnerable in a particular operational environment and how this environment could be affected in case that the device is compromised or insecure.

Modularity as a Key Concept

This guide is not a monolithic, all-encompassing instruction manual for IoT device penetration testing. Instead, it should be seen as a dynamic and growing collection of test cases for various technologies related to IoT devices.

In its current state, this guide comprises test cases on a very high and generic level. This is intentional since the base version of this guide should be applicable to as many different IoT devices as possible (*comparability*). However, the long-term goal is that this guide will be expanded over time by adding modules with more detailed test cases for specific technologies (*expandability*). Thereby, the guide will evolve and become more and more detailed over time.

Solution Approach

During the preparation of a penetration test, a series of important decisions need to be made, which have a major impact on the test procedure and consequently the test results. Part of these decisions is to clarify what should be tested (*scope of the test*) and how the test should be performed (*test perspective*).

In order to achieve the proposed solution, the following approach was chosen:

1. **Creation of an IoT device model, which represents an abstract, generalized IoT device:**

   Before the test scope for an IoT device penetration test can be identified, it must first be defined what an IoT device is and which parts it consists of. In order to support the test scope definition, the device model should include device components that can either be included in or excluded from the test scope. This guide will only focus on components, directly belonging to the device itself. All device-external elements, such as web applications, mobile applications and back end servers, will not be part of this guide although sister OWASP testing guides cover these areas. The device model will serve as a generalized scheme, depicting the common structure of IoT devices, thereby enhancing the comprehensibility and comparability of the methodology presented in this guide. As all further parts of the guide will rely on this basis, the creation of the device model is a mandatory and important first step.

2. **Creation of an attacker model, which represents and categorizes potential attackers:**

   The guide will comprise key aspects of testing for each component of the device model. Therefore, it will include a catalog of potential test cases for all device components. Since it might not be required to perform all of these test cases for any given kind of IoT device, a systematic approach is required, which yields a selection of applicable test cases based on the requirements and the intended operational environment of a specific device. The attacker model will support the definition of the test perspective, providing comprehensibility and comparability by defining common groups/types of attackers. In order to maintain efficiency, the attacker model will not incorporate extensive threat and risk analysis models. This also benefits the comparability across different device implementations.

3. **Creation of a test methodology, which includes general key aspects of testing:**

   Based on the IoT device model, a testing methodology including general key aspects of testing will be developed. These general key aspects represent security issues that are relevant for the device components and will be derived from more detailed test cases for specific exemplars of a given component or technology. This derivation should decouple the key aspects from specifics of the exemplar in order to enable the methodology to be used for as many different IoT device implementations as possible (*comparability*). However, the structure of the methodology will allow to add more detailed key aspects of testing for specific exemplars of a device component later on, thus providing expandability.

2. IoT Security Testing Framework

2.1. IoT Device Model

This chapter will focus on the IoT device model representing the general structure of IoT devices. Creating the device model is the first step in order to achieve the goals defined in the solution approach (see 1. Introduction). All further steps, which will be described in 2.2. Attacker Model, 2.3. Methodology and 3. Test Case Catalog, will be based on the device model.

Related Work

The device model was built upon a reference architecture for IoT platforms. Furthermore, potential attack vectors in the form of attack surface areas were also taken into account since the device model will be used in a security context. These are outlined by the following related work:

- **"Comparison of IoT Platform Architectures: A Field Study based on a Reference Architecture":** The aim of this paper is to propose a reference architecture for IoT ecosystems. This reference architecture was "kept [...] abstract on purpose since the aim of [the] reference architecture is to serve as a uniform, abstract terminology, which eases the comparison of different platforms" (source). As the device model developed in this guide should also serve as a uniform model for different IoT devices, independent of specific implementations and designs, the reference model after Guth et al. (source) was taken as a basis. Nevertheless, the model after Guth et al. (source) is superficial in terms of the IoT device itself. It depicts the device as a single component without further differentiation of its parts (besides drivers). Thus, it is not sufficient for this guide since it does not allow a fine-grained definition of the test scope (inclusion and exclusion of specific device parts). In the model introduced in this guide some adjustments were made in order to further differentiate individual parts of IoT devices.
- **"IoT Attack Surface Areas Project":** OWASP regularly publishes penetration testing methodologies and collections of popular security risks (called "OWASP Top 10") in several technical fields, such as web and mobile application security. Due to its popularity, it has become one of the major sources for information regarding penetration testing. In 2014 and 2018, OWASP has also published a top 10 of security risks regarding the IoT field. The surface areas mentioned in the "IoT Attack Surface Areas Project" represent parts of an IoT solution which might be targeted by potential attackers. Due to the fact that this list already covers many potential attack vectors in regards to IoT devices and IoT ecosystems in general, it was also used as a basis for the device model proposed within this guide. However, some adjustments were made in order to further differentiate the details of IoT device implementations, especially in terms of the hardware side. Furthermore, the "IoT Attack Surface Areas Project" only consists of a simple list of device parts, which does not specify how these parts interact

with each other. It also misses to define the characteristics of each device part (or respectively the attack surface area) and thus makes it difficult to differentiate them, e.g., "Device Memory" and "Local Data Storage". (source)

Device Boundaries

In order to distinguish between components belonging to an IoT device and components of the surrounding IoT ecosystem, it is necessary to first define the boundaries of an IoT device. An IoT device is generally encompassed by an enclosure of some kind, which (physically) separates device-internal lements from device-external elements.

Interactions between internal and external elements are only possible via interfaces. Within this guide, these interfaces are not considered to be part of the enclosure. Instead, those interfaces will be categorized individually (see Interfaces).

As will be explained in the next section, the term "component" refers to an item that can be the subject of a penetration test. Thus, device-internal elements and interfaces are considered components within this guide.

Components

As introduced in the previous sections, the proposed device model should provide a generalized selection of parts that IoT devices consist of. These parts will be referred to as components. Every component is a piece of soft- and/or hardware that, in theory, can be tested individually. The penetration test scope for an IoT device can therefore be defined as a list of components.

Device-Internal Elements

Every device-internal element is a component residing inside the device enclosure. Thus, they are part of the IoT device. IoT devices usually comprise the following internal elements, all of which are mentioned in the list of attack surfaces composed by OWASP (source):

- **Processing unit:** The processing unit, also called processor, is responsible for managing and performing data processing tasks. These tasks are defined as a sequence of instructions that are loaded from the memory. A device has at least a central processing unit handling its core functionalities (defined by the firmware). However, more complex devices might also be equipped with further processing units that are assigned to specific subtasks. A special kind of processor are microprocessors, built on a single circuit. Microcontrollers are microprocessors, which also have analog and digital in- and outputs. They are typically used to control the behavior of a device and are often used in the embedded field. (source)

Examples: x86 processor, ARM processor, AVR processor

- **Memory:** Memory is used to store data, such as programs (instructions for a processing unit) and information, in binary form. Depending on the type of memory, it is used to temporarily store data while being processed by a processing unit (primary memory or cache) or to permanently store data on a device even while the device is turned off (secondary memory). A special kind of secondary memory is flash memory. It is commonly used in many devices because it is energy-saving, develops less heat and is less susceptible to vibration and magnetic fields due to the lack of moving parts. Flash memory is based on semiconductor technology and able to provide fast and permanent access to data (read, write, delete). (source, source)

  Examples: EEPROM, flash memory

- **Firmware:** "Firmware is a software program or set of instructions programmed on a hardware device" (source). It is used to control the device and the communication between device-internal and -external elements (data in- and output via data exchange services). Firmware is stored on a memory and executed by a processing unit. In regards of device firmware, the following components might be potential targets for a penetration test:

  - **Installed firmware:** Installed firmware refers to firmware that is already installed on a device. It might be the target of dynamic analyses and usually handles the storage and processing of sensitive user data.

  - **Firmware update mechanism:** A firmware update mechanism is part of the firmware and defines how firmware updates, in the form of firmware packages, can be installed on a device. A crucial responsibility of a firmware update process is to ensure that only proper firmware packages can be installed and executed.[1]

  Examples: OS, RTOS, bare-metal embedded firmware

- **Data exchange service:** Data exchange services refer to programs or parts of programs, used to transfer data between two or more components via an interface (e.g., network, bus). These services are part of the firmware and can be used to transmit data, receive data or both.

  Examples: network service, debug service, bus listener

[1] For performing a test of a firmware update mechanism, a firmware package is required. Due to the fact that a firmware package could also be inspected separately, it could be considered a component as well. However, since this guide focuses on device-internal elements and device interfaces only, firmware packages are not in scope. Contrary to installed firmware, an update package also includes the firmware header, which might include important data.

Interfaces

Interfaces are required to connect two or more components with each other. Interactions between device-internal elements or between device-internal and device-external elements are only possible via interfaces. Based on which components are connected by an interface, it can be categorized as a machine-to-machine or human-to-machine interface. As long as at least one of the connected components is a device-internal element, the interface itself is also part of the device.

Within this guide, the following kinds of interfaces will be differentiated, all of which are either directly or indirectly mentioned in the list of attack surfaces, composed by OWASP (source):

- **Internal interfaces (machine-to-machine):** These interfaces are used to establish a connection between device-internal elements and are not accessible from outside the device enclosure.

  Examples: JTAG, UART, SPI

- **Physical interfaces (machine-to-machine):** Physical interfaces are used to establish a connection between device-internal and -external elements, based on a physical connection between the components or the respective interfaces of those components. Therefore, physical interfaces require a socket or a port, built into the device enclosure and thus are accessible from outside the device.

  Examples: USB, Ethernet

- **Wireless interfaces (machine-to-machine):** Similar to physical interfaces, wireless interfaces are also used to establish a connection between device-internal and -external elements. However, the connection between wireless interfaces is not based on a physical connection, but on radio waves, optical signals or other wireless technologies. Wireless interfaces are accessible from outside the device, usually from a greater distance than physical interfaces.

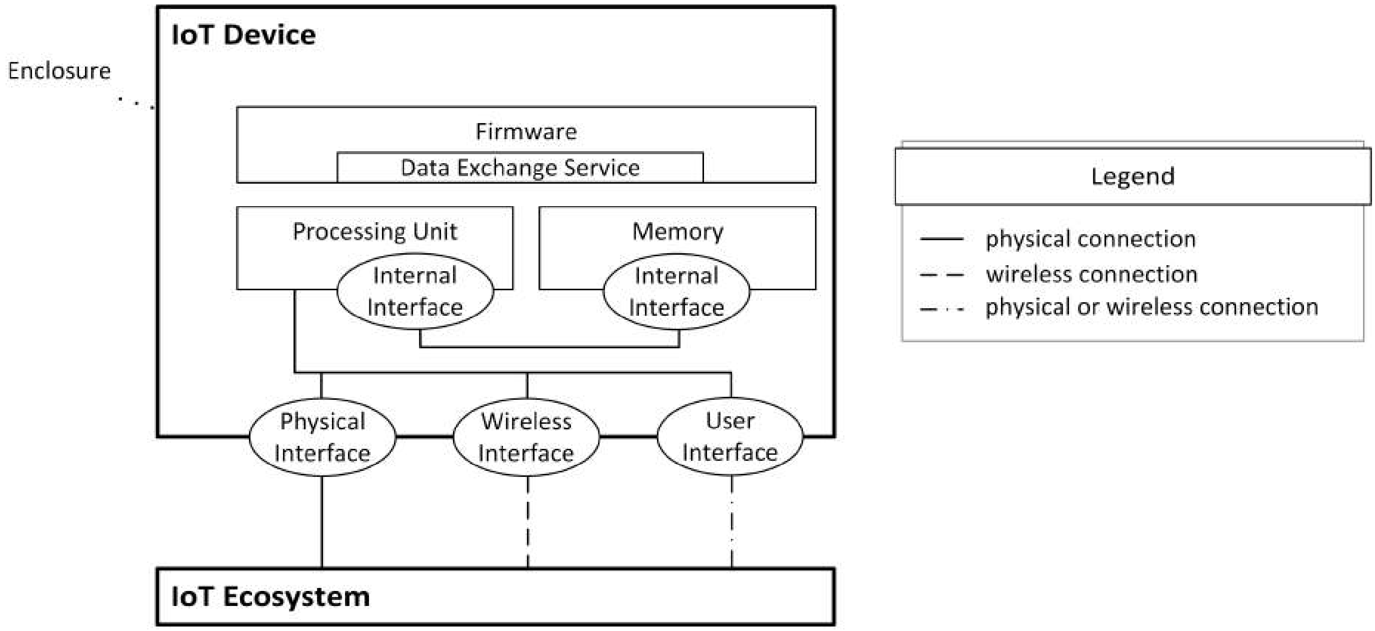
  Examples: Wi-Fi, Bluetooth, BLE, ZigBee

- **User interfaces (human-to-machine):** In contrast to all other above-mentioned interfaces, user interfaces are not utilized to establish a connection between two machines. Instead, their purpose is to allow interactions between device-internal elements and a user. These interactions can either be based on a physical connection, e.g., in case of a touch display, or wireless connections, e.g., in case of a camera or microphone.

  Examples: touch display, camera, microphone, local web application (hosted on the device)

Device Model Scheme

The device model is a combination of all above-mentioned components and can be seen in the figure below. It must be noted that, even though cardinalities were not included for better readability, more than one instance of each component might be built into an IoT device.



Other models, e.g., the ones mentioned in Related Work, include sensors and actors as components of a device. Within this guide, sensors and actors are considered physical, wireless or user interfaces respectively because they enable interactions between device-internal and -external elements or users via physical (e.g., touch sensor, door control) or wireless connections (e.g., microphone, temperature sensor).

In some cases, it is also possible that devices comprise parts which can be considered devices themselves (i.e., nested devices). It then depends on the perspective of the observer which interfaces are classified as internal and external. The determining factor are the boundaries between the observer and the interface (see Device Boundaries, Device-Internal Elements and Interfaces).

Overall, the device model, which was specifically developed in the context of this guide, can be used to create and share abstract representations of various different IoT devices. Contrary to other models, this one solely focuses on the IoT device and the components it is built of. Hence, the model allows to describe device implementations in a more detailed manner. In combination with the models and concepts, developed in the following chapters, it is possible to compile a list of applicable test cases for any given device regardless of the specific technologies or standards that are implemented.

2.2. Attacker Model

In this chapter, a selection scheme for test cases will be described, which is based on potential attackers that are assumed to be a threat to a given IoT device. Contrary to a full threat and risk modeling approach, like the STRIDE model, the attacker model used in this guide presents a more streamlined procedure for defining and selecting threats to IoT devices.

The reasons for not using a formal threat and risk modeling approach are:

- Threat and risk modeling is usually focused on one specific implementation design. Thus, the identified threats and risks are based on certain conditions of a given solution or device, which makes it difficult to compare different solutions with each other.

- Performing a formal threat and risk analysis requires a significant amount of time, which further increases with the complexity of the subject. Making a formal threat and risk analysis a mandatory requirement for penetration tests would result in longer testing periods and consequently higher expenses per test.

The spectrum of potential attackers reaches from anonymous global attackers to privileged individuals and users of the device. As will be explained in following sections, the list of attackers can be narrowed down by defining minimum and maximum access requirements, representing the test perspective. Every device component and test case will be tagged with the access level, which is required to perform the respective tests. Hence, the list of device components in scope of the test as well as the list of applicable test cases will be a result of applying the attacker model on the results, yielded by the device model.

It must be noted that, within this chapter, the term "IoT device" refers to a single device or device type whereas, in the other chapters of this guide, it refers to IoT devices in general.

Conceptual Basis for the Attacker Model

This attacker model will characterize groups of potential attackers based on their access capabilities[1]. The metrics that are used for this attacker model are based on the metrics of the CVSS. Even though the CVSS is primarily used to rate the severity of vulnerabilities in the web application and computer networking field, it implements a straightforward approach to assess the capabilities of attackers and the conditions that are required to exploit certain security issues. Another benefit of using a model that is similar to the CVSS is that many security professionals are already working with the CVSS. Hence, many testers and manufacturers/operators are familiar with this system, which also contributes to the acceptance of this attacker model.

The CVSS defines the following exploitability metrics:

- **Attack vector:** "This metric reflects the context by which vulnerability exploitation is possible" (source). Values for this metric are ranging from network access (e.g., via the internet) to physical access. Within the attacker model, this metric will be reflected by the physical access level.
- **Attack complexity:** "This metric describes the conditions beyond the attacker's control that must exist in order to exploit the vulnerability" (source). The attack complexity is not used in the attacker model since it refers to "conditions beyond the attacker's control" (source) and thus is not relevant for categorizing potential attackers.
- **Privileges required:** "This metric describes the level of privileges an attacker must possess before successfully exploiting the vulnerability" (source). Values for this metric are ranging from none (no privileges) to high. The required privileges are represented in the attacker model by the authorization access level.
- **User interaction:** "This metric captures the requirement for a human user, other than the attacker, to participate in the successful compromise of the vulnerable component" (source). The necessity of interactions by legitimate users will not be considered in the attacker model since, while being relevant for the exploitability of a vulnerability, it is not relevant for the selection of applicable test cases.

[1] In regards of IT security, attackers are usually characterized based on further factors, e.g., their aggressiveness and their resources (processing power, time, money). However, these factors do not have or only have a minor impact for the selection of applicable test cases.

Access Levels

Within this attacker model, access levels are a measure for the relation between a certain group of individuals (access group) and the IoT device. They describe how individuals of the access group are intended to be able to interact with the device. These can either be physical interactions or logical authorization interactions.

The degree of how close individuals can get to the device is measured by the physical access level. The physical access level is an adaption of the CVSS metric "attack vector" and it reflects the physical context that is required to perform attacks against a target device. Therefore, some of the original values from the CVSS were used (network, local, physical). However, the description of local access was adjusted in regards of the focus on the physical context. Additionally, the physical access as defined in the CVSS was split into two levels: non-invasive and invasive physical access. The reason for this is that some IoT devices are protected with special measures that restrict access to device-internal elements, e.g., locked or sealed enclosures. In this case, attackers might not be able to access device-internals in a reasonable amount of time, thus they only have non-invasive physical access. Other devices have enclosures that can be opened in a short time, e.g., by removing screws. Thus, attackers could access device-internals, therefore gaining invasive physical access. Overall, the physical access level can be affected by factors like geographical location, building security or the device enclosure.

The following physical access levels are defined:

1. **Remote access (*PA-1*):** There is an arbitrary physical distance between an individual and the device. An attacker with remote access can be located anywhere in the world, which usually means that the device is directly accessible via a Global Area Network (GAN).

2. **Local access (*PA-2*):** There is a limited physical distance[2] between an individual and the device, but direct physical interactions are not possible. An attacker with local access can use the device from close proximity, which usually means that the device is directly accessible via a Local Area Network (LAN) or Wireless Local Area Network (WLAN).

3. **Non-invasive access (*PA-3*):** There is no physical distance between an individual and the device, but the individual cannot directly access device-internal elements in a physical manner (i.e., cannot easily open the device enclosure).

4. **Invasive access (*PA-4*):** There is no physical distance between an individual and the device and the individual can directly access device-internal elements in a physical manner (i.e., open the device enclosure).

The digital privileges of individuals are measured by the authorization access level. The authorization access level is an adaption of the CVSS metric "privileges required". In addition to the values, defined in the CVSS, another level of privileges, called manufacturer-level access, was added on top of the high privileges. Contrary to web applications and computer networks, which are usually operated from within the control zone of the operator (e.g., within a data center), IoT devices are often operated outside that control zone. Established methods for securing maintenance and debugging access (e.g., restricting maintenance access to pre-defined subnets, IP addresses or physical ports in the data center) can not always be applied. Hence, attacks against a device with manufacturer-level access might be possible. Overall, the authorization access level can be affected by factors like policies or role-based access models.

The following authorization access levels are defined:

1. **Unauthorized access (*AA-1*):** An individual can get anonymous access to the device component. Attackers with anonymous access can be any unregistered user.

2. **Low-privileged access (*AA-2*):** An individual can only get access to the device component, if it is authenticated and in possession of standard authorization privileges. Attackers with low-privileged access can be any registered user.

3. **High-privileged access (*AA-3*):** An individual can only get access to the device component, if it is authenticated and in possession of extensive privileges. The term

"extensive privileges" means that individuals have access to restricted functionalities that are not available to all registered users of the device component (e.g., configuration settings).

4. **Manufacturer-level access (*AA-4*):** An individual can only get access to the device component, if it is authenticated and in possession of manufacturer-level authorization privileges. Contrary to high-privileged access, manufacturer-level access is not restricted in any way and includes, e.g., debugging access for developers of the device, access to the source code or root-level access to the firmware.

[2] Limited physical distance is not restricted to a specic maximum value per se. Depending on the technologies in use, the maximum distance might range from a few meters (e.g., in case of Bluetooth) to a few kilometers (e.g., in case of LTE).

Mapping of Device Components and Access Levels

The perspective of the testers during the test will be determined by minimal and maximal access levels, chosen as a baseline for the test. Physical and authorization access levels have different impacts on the penetration test and its scope.

Physical access level:

- The physical access level refers to the device as a whole. Thus, some physical access levels directly define that certain device components can not be tested with the given level since an attacker could not interact with these components at all. The relation between physical access levels and device components is shown in the table below.
- Based on the specific requirements of a manufacturer or operator, the minimal and/or maximal physical access levels might be hard boundaries for the test execution since the contractee might want to specifically exclude certain tests, e.g., those which require invasive physical access.

Authorization access level:

- Since authorization access might be handled differently across multiple device components, the authorization access level rather refers to access to an individual component than to the device as a whole. Thus, the impact of authorization access levels on the test scope always depends on the specific implementation of the business logic and the authorization/permission scheme per component.
- There is no reason for selecting a minimal authorization access level for the test perspective since evaluating whether it is possible to get access to (parts of) the device with lower privileges than intended should be part of the test.

All in all, the attacker model can be used to create an abstract representation of potential attackers. It can be used to describe which kind of attackers is considered a threat to a given device in its operation environment. Contrary to other methodologies and models, this one can be used in a more streamlined manner, thus being more efficient, e.g., compared to full threat and risk analysis approaches. It is also takes the specifics of the IoT context more into account than the CVSS, which it is based on. In combination with the device model, it is possible to define the test scope and test perspective, thereby determining which test cases can and shall be performed.

Component	**PA-4**	**PA-3**	**PA-2**	**PA-1**
Processing Unit	✓			
Memory	✓			
Installed Firmware	✓	**?**[3]	**?**[3]	**?**[3]
Firmware Update Mechanism	✓	**?**[3]	**?**[3]	**?**[3]
Data Exchange Service	✓	**?**[4]	**?**[4]	**?**[4]
Internal Interface	✓			
Physical Interface	✓	✓	**?**[5]	
Wireless Interface	✓	✓	✓	
User Interface	✓	✓	**?**[6]	**?**[6]

[3] Installed firmware and the firmware update mechanism might be testable with non-invasive (*PA-3*), local (*PA-2*) or remote physical access (*PA-1*), depending on how direct access to the firmware can be accomplished (e.g., via SSH).

[4] Data exchange services might be testable with non-invasive (*PA-3*), local (*PA-2*) or remote physical access (*PA-1*), depending on if they were designed for that kind of access, e.g., for remote control or monitoring purposes.

[5] Physical interfaces might be testable with local physical access (*PA-2*) under certain circumstances, e.g., if the physical interface is connected to a local network.

[6] User interfaces might be testable with local (*PA-2*) or remote physical access (*PA-1*), depending on if they were designed for that kind of access, e.g., for remote control or monitoring purposes.

2.3. Testing Methodology

In this chapter, a methodology for performing IoT device penetration tests will be described. It is based on the concepts, presented in 2.1. IoT Device Model and 2.2. Attacker Model and serves as a supplement, which can be used with pre-existing penetration testing workflows and frameworks. The methodology comprises key aspects of testing that have to be performed during an IoT device penetration test. Therefore, it includes a catalog of test cases for each individual device component. As described in the previous chapters, the specific selection of applicable test cases depends on the results of applying the device and attacker models, which have been designed in the context of this methodology.

At first, it will be described how this methodology can be integrated into other workflows and during which steps the models and concepts of this methodology can be used. Then, selected testing techniques will be explained, which can be applied during the test and are not restricted to certain test cases. Finally, the structural concept of the catalog of test cases will be explained.

In comparison to other IoT penetration testing frameworks, this methodology follows a more generic yet comprehensive approach. It defines test cases for certain security issues that are relevant in the IoT context (key aspects of testing) without being restricted by the details of specific technologies or standards. Thereby, this methodology is more flexible than other frameworks, which is an important benefit given the volatility of the IoT field. Nonetheless, the methodology is applicable to various technologies and provides possibilities for further particularizations.

It must be noted that test cases, which apply to multiple components, will not be included in this chapter. The full list of test cases can be found in 3. Test Case Catalog.

Integration Into Other Workflows and Frameworks

To achieve efficiency, no major adjustments to any pre-existing workflows should be required to incorporate this methodology. In the following, it will be shown how this methodology can be integrated into other frameworks based on the example of the BSI penetration testing model (source). In this case, no changes to the overall test workflow are required.

The methodology proposed in this guide can be used to facilitate the following steps:

- **Clarification of the Test Scope and Test Perspective:** The methodology supports the clarification of test objectives and conditions with the contractee during phases 1 and 3 of the BSI model (source) by establishing a common terminology in form of the device and attacker models, thus facilitating the communication. Furthermore, the device model supports the testing team during phase 2 by providing a generic scheme, which can be compared to the architecture of a given IoT device in order to identify potential attack vectors.

- **Test Execution and Documentation:** The catalog of test cases acts as a guideline for testers during the active test (phase 4 of the BSI model (source)). Depending on the test scope and test perspective, applicable test cases are defined, which have to be performed during the test. Thus, the test catalog can be used as a checklist to ensure that all mandatory tests were performed. It also allows to transparently document the test procedure in a reproducible manner during phase 5, due to the fact that performed test cases can be referenced in the report.

Description of the Hierarchic Structure

In the following, the overall structure of the test case catalog as well as the general layout of a test case will be defined.

Structure of the Catalog of Test Cases

The catalog of test cases will follow a hierarchic (tree) structure. Starting from a single root node (IOT), each component of the device model will be represented as a child node, thereby forming its own subtree. Subsequently, further nodes will be added as children to the component nodes, eventually resulting in each test case being a leaf node. A unique identifier, incorporating this structure, will be assigned to each node, allowing to reference it in the test report or other documents.

The following hierarchic levels and types of nodes are defined:

- **Component:** The first main hierarchy level is the component (see 2.1. IoT Device Model). The type of component (device-internal element/interface) was not included in the hierarchy for the sake of simplicity and due to the lack of added value.

  Short representation: 2 - 5 uppercase alphabetic characters

  Examples: ISTG-PROC, ISTG-MEM, ISTG-FW, ISTG-DES, ISTG-INT, ISTG-PHY, ISTG-WRLS, ISTG-UI

- **Component Specialization (Optional):** Optional component specializations can be used to define test cases that are only relevant for certain parts or exemplars of a component (e.g., installed firmware - ISTG-FW[INST] - as specialization for the component firmware - ISTG-FW - or SPI - ISTG-INT[SPI] - as specialization for the component internal interface - ISTG-INT).

  By default, component specializations inherit all categories and test cases, defined for their parent node (e.g., all test cases defined for the component firmware - ISTG-FW -

are inherited by the specialization installed firmware - ISTG-FW[INST]).

If required, it is allowed to chain specializations, for example over-the-air firmware updates - ISTG-FW[UPDT][OTA] - as specialization of firmware update - ISTG-FW[UPDT]. In this case, the second specialization inherits all categories and test cases, defined for the first specialization, thus also inheriting all test cases, defined for the component in general.

Furthermore, if required, it is also allowed to define a list of categories or test cases, which should be excluded from being inherited by a component specialization.

Short representation: 2 - 5 uppercase alphabetic characters in square brackets

Examples: ISTG-FW[INST], ISTG-FW[UPDT]

- **Category:** The second main hierarchy level is the category, which can be used to group test cases, e.g., all test cases related to authorization can be grouped in the category AUTHZ.

  Short representation: 2 - 5 uppercase alphabetic characters

  Examples: ISTG--AUTHZ, ISTG-*-INFO, ISTG-*-CONF*

- **Test Case:** The third main hierarchy level is the test case. See 3. Test Case Catalog for more details.

  Short representation: three-digit incremental number of the test case.

  Examples: ISTG-FW-INFO-001, ISTG-FW-INFO-002, ISTG-FW-INFO-003

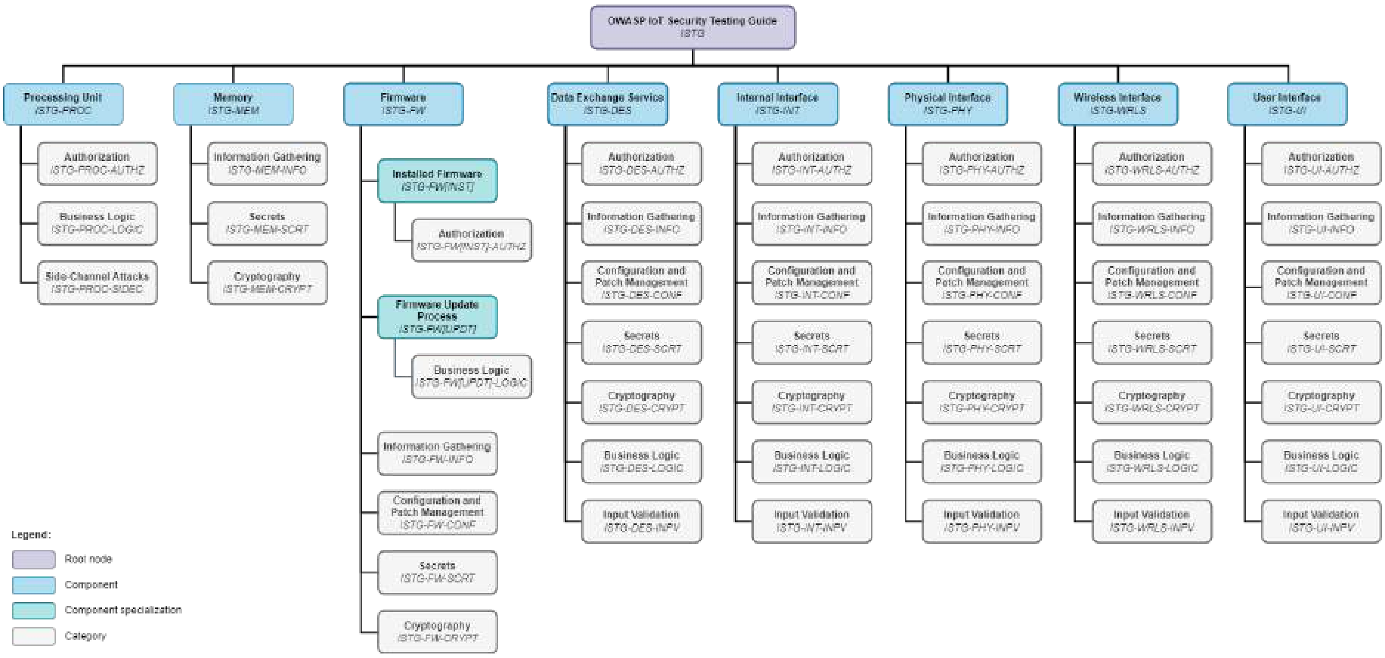
This kind of structure allows to efficiently determine applicable subtrees by deselecting nodes (e.g., components, component specializations and categories) that are not relevant for a given device or test scenario. The table below shows an exemplary list of nodes for each hierarchy level. An overview of all components and categories that are included in this guide can be seen in the figure below the table.

The usage of component and category specializations allows to expand the catalog of general test cases to include test cases for specific standards and technologies. By inheriting test cases from their parent nodes, it is ensured that these test cases are also applied to the child nodes by default. However, at the time of writing this guide, the possibility that test cases of a parent node might not be applicable to a child node in particular cases could not be precluded. Thus, it is allowed to specify a list of test cases, which are excluded from being inherited by a certain child node.

Another way to expand the catalog is to add custom components, categories and test cases. This way, the methodology could also be expanded to include further components, e.g., device-external elements of the IoT ecosystem.

Hierarchy Level	ID	Description
0	IOT	Root Node
1	**Component**	
	ISTG-PROC	Processing Unit
	ISTG-MEM	Memory
	ISTG-FW	Firmware
	ISTG-DES	Data Exchange Service
	ISTG-INT	Internal Interface
	ISTG-PHY	Physical Interface
	ISTG-WRLS	Wireless Interface
	ISTG-UI	User Interface
	ISTG-*	Custom Component *(placeholder for future extensions)*
	Component Specialization (Optional)	
	ISTG-FW[INST]	Installed Firmware
	ISTG-FW[UPDT]	Firmware Update Mechanism
	ISTG-*[*]	Custom Component Specialization *(placeholder for future extensions)*
2	**Category**	
	ISTG-*-AUTHZ	Authorization
	ISTG-*-INFO	Information Gathering
	ISTG-*-CRYPT	Cryptography
	ISTG-*-SCRT	Secrets
	ISTG-*-CONF	Configuration and Patch Management
	ISTG-*-LOGIC	Business Logic
	ISTG-*-INPV	Input Validation
	ISTG-*-SIDEC	Side-Channel Attacks
	ISTG-*-*	Custom Category *(placeholder for future extensions)*
3	**Test Case**	
	ISTG-*-INFO-001	Disclosure of Source Code and Binaries
	ISTG-*-INFO-002	Disclosure of Implementation Details

Hierarchy Level	ID	Description
	ISTG-*-INFO-003	Disclosure of Ecosystem Details
	ISTG-*-*-*	Custom Test Case *(placeholder for future extensions)*



Structure of Test Cases

Each individual test case, which is represented by a leaf node, is divided into the following sections:

- **Requirements:** The requirements section will define which physical and authorization access levels are required to carry out the test case. Since these requirements also depend on the given test conditions, e.g., the specific implementation of the target device and its operational environment, a range of access levels might be defined which apply to the test case in general.
- **Summary:** The summary section includes an overall description of the security issue, which the test case is based on.
- **Test Objectives:** In the test objectives section, a list of checks that the tester has to perform is given. By performing these checks, the tester can determine whether the device is affected by the security issue described in the summary.
- **Remediation:** The remediation section comprises recommendations regarding potential measures that can be applied to solve the security issue. However, these recommendations are only rough suggestions. It is the responsibility of the

manufacturer/operator to derive detailed measures in regards of the device implementation.

3. Test Case Catalog

3.1. Processing Units (ISTG-PROC)

Table of Contents

Overview

This section includes test cases and categories for the component processing unit. A processing unit is a device-internal element that can only be accessed with *PA-4*. Establishing a direct connection to the processing unit might require specific hardware equipment (e.g., a debugging board, an oscilloscope or test probes).

The following test case categories, relevant for processing units, were identified:

- **Authorization:** Focuses on vulnerabilities that allow to get unauthorized access to the processing unit or to elevate privileges in order to access restricted functionalities.

- **Business Logic:** Focuses on vulnerabilities in the design and implementation of instructions as well as the presence of undocumented, potentially vulnerable, instructions.

- **Side-channel Attacks:** Focuses on the resilience against side-channel attacks like timing and glitching attacks.

Authorization (ISTG-PROC-AUTHZ)

Depending on the access model for a given device, only certain individuals might be allowed to access a processing unit directly. Thus, proper authentication and authorization procedures need to be in place, which ensure that only authorized entities can get access.

Unauthorized Access to the Processing Unit (ISTG-PROC-AUTHZ-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1*

Summary

Depending on the specific implementation of a given device, access to a processing unit might be restricted to entities with a certain authorization access level, e.g., *AA-2*, *AA-3* or *AA-4*. If the device fails to correctly verify access permissions, any attacker (*AA-1*) might be able to get access.

Test Objectives

- It must be checked if authorization checks for access to the processing unit are implemented.
- In case that authorization checks are in place, it must be determined whether there is a way to bypass them.

Remediation

Proper authorization checks need to be implemented, which ensure that access to the processing unit is only possible for authorized entities.

References

This test case is based on: ISTG-DES-AUTHZ-001.

Privilege Escalation (ISTG-PROC-AUTHZ-002)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-2* - *AA-3* (depending on the access model for the given device)

Summary

Depending on the specific implementation of a given device, access to some functionalities of a processing unit might be restricted to individuals with a certain authorization access level, e.g., *AA-3* or *AA-4*. If the processing unit fails to correctly verify access permissions, an attacker with a lower authorization access level than intended might be able to get access to the restricted functionalities.

Test Objectives

- Based on ISTG-PROC-AUTHZ-001, it must be determined whether there is a way to elevate the given access privileges and thus to access restricted functionalities.

Remediation

Proper authorization checks need to be implemented, which ensure that access to restricted functionalities is only possible for individuals with the required authorization access levels.

References

This test case is based on: ISTG-DES-AUTHZ-002.

Business Logic (ISTG-PROC-LOGIC)

Issues in the underlying logic of a processing unit might render the device vulnerable to attacks. Thus, it must be verified if the processing unit and its functionalities are working as intended and if exceptions are detected and properly handled.

Insecure Implementation of Instructions (ISTG-PROC-LOGIC-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on what level of privileges is required to successfully submit ins

Summary

Flaws in the implementation of the business logic might result in unintended behavior or malfunctions of the device. For example, if an attacker intentionally tries to skip or change important instructions in the processing workflow, the device might end up in an unknown, potentially insecure state.

Test Objectives

- Based on the specific implementation, it has to be determined whether instructions can be misused to manipulate the behavior of the device.

- It must be checked if the processing unit in use supports undocumented, potentially vulnerable instructions. For example, this can be done by fuzzing instructions or performing research regarding the processing unit model.

Remediation

The device should not end up in an unknown state. Anomalies in the workflow must be detected and exceptions have to be handled properly.

References

This test case is based on: ISTG-DES-LOGIC-001.

Side-Channel Attacks (ISTG-PROC-SIDEC)

Side-channel attacks, such as timing and glitching attacks, are usually targeted against the physical implementation of a device or more specifically a processing unit instead of the device firmware or its interfaces. The goal of such attacks is to gather information about cryptographic algorithms and operations, performed by a processing unit, in order to retrieve key material, manipulate the cryptographic calculations or gain access to protected information.

Insufficient Protection Against Side-Channel Attacks (ISTG-PROC-SIDEC-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on how the attack is being performed; see summary for more c

Summary

As mentioned above, side-channel attacks can be used by an attacker to get access to sensitive data or to manipulate the device operation. Usually, side-channel attacks are customized attacks tailored to a specific hardware implementation.

Depending on how the attack is being performed, different levels of authorization access might be required. Some side-channel attacks, such as glitching attacks, do not require authorization access at all since the attack is performed on a physical level by manipulating the power supply. Other side-channel attack vectors, such as the Meltdown vulnerability, require the execution of code by an attacker. Thus, some kind of authorization access is necessary.

Test Objectives

- It has to be determined whether the processing unit is affected by known vulnerabilities, such as Meltdown and Spectre.
- During the testing period, the behavior of the processing unit has to be analyzed in order to assess the probability of successful side-channel attacks like timing or

glitching attacks.

Remediation

Based on the results of the analysis, the hardware design should be adjusted to be resilient against side-channel attacks. Furthermore, if publicly known vulnerabilities exist, the latest patches should be installed.

References

For this test case, data from the following sources was consolidated:

- "A practical implementation of the timing attack" by Jean-François Dhem, François Koeune, Philippe-Alexandre Leroux, Patrick Mestré, Jean-Jacques Quisquater, and Jean-Louis Willems *(In Jean-Jacques Quisquater and Bruce Schneier, editors, Smart Card Research and Applications, pages 167 - 182, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.)*
- "Injecting Power Attacks with Voltage Glitching and Generation of Clock Attacks for Testing Fault Injection Attacks" by Shaminder Kaur, Balwinder Singh, Harsimranjit Kaur, and Lipika Gupta *(In Pradeep Kumar Singh, Arti Noor, Maheshkumar H. Kolekar, Sudeep Tanwar, Raj K. Bhatnagar, and Shaweta Khanna, editors, Evolving Technologies for Computing, Communication and Smart World, pages 23 - 37, Singapore, 2021. Springer Singapore.)*
- "Spectre attacks: Exploiting speculative execution" by Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom *(In 40th IEEE Symposium on Security and Privacy (S&P'19), 2019.)*
- "Meltdown: Reading kernel memory from user space" Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg *(In 27th USENIX Security Symposium (USENIX Security 18), 2018.)*

3.2. Memory (ISTG-MEM)

Table of Contents

Overview

This section includes test cases and categories for the component memory. Similar to the processing unit, the memory is a device-internal element that can only be accessed with *PA-4*. Establishing a direct connection to the memory might require specific hardware equipment (e.g., a debugging board or test probes).

In regards to test case categories that are relevant for memory, the following were identified:

- **Information Gathering:** Focuses on information that is stored on the memory chip and that might be disclosed to potential attackers if not being properly protected or removed.

- **Secrets:** Focuses on secrets that are stored on the memory chip in an insecure manner.

- **Cryptography:** Focuses on vulnerabilities in the cryptographic implementation.

Information Gathering (ISTG-MEM-INFO)

The memory of an IoT device can include various data, which, if disclosed, could reveal details regarding the inner workings of the device or the underlying IoT ecosystem to potential attackers. This could enable and facilitate further, more advanced attacks.

Tests on the device memory are performed by directly accessing the memory chips. Thus, invasive physical access (*PA-4*) is required while no user accounts are used (*AA-1*).

Disclosure of Source Code and Binaries (ISTG-MEM-INFO-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1*

Summary

The disclosure of uncompiled source code could accelerate the exploitation of the software implementation since vulnerabilities can be directly identified in the code without the need to perform tests in a trial and error manner. Furthermore, left-over source code might include internal development information, developer comments or hard-coded sensitive data, which were not intended for productive use.

Similar to uncompiled source code, compiled binaries might also disclose relevant information. However, reverse-engineering might be required to retrieve useful data, which could take a considerable amount of time. Thus, the tester has to assess which binaries might be worth analyzing, ideally in coordination with the device manufacturer.

Test Objectives

- It must be checked if uncompiled source code can be identified within the device memory.
- If uncompiled source code is detected, its content must be analyzed for the presence of sensitive data, which might be useful for potential attackers.
- Reverse-engineering of selected binaries should be performed in order to obtain useful information regarding the device implementation and the processing of sensitive data.

Remediation

If possible, uncompiled source code should be removed from devices, intended for productive use. If the source code has to be included, it must be verified that all internal development data is removed before releasing the device.

Since it is not possible to prevent reverse-engineering completely, measures to restrict access to the device memory in general should be implemented to reduce the attack surface. Furthermore, the reverse-engineering process can be impeded, e.g., by obfuscating the code.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta

This test case is based on: ISTG-FW-INFO-001.

Disclosure of Implementation Details (ISTG-MEM-INFO-002)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1*

Summary

If details about the implementation, e.g., algorithms in use or the authentication procedure, are available to potential attackers, flaws and entry points for successful attacks are easier to detect. While the disclosure of such details alone is not considered to be a vulnerability, it facilitates the identification of potential attack vectors, thus allowing an attacker to exploit insecure implementations faster.

Test Objectives

- Accessible details regarding the implementation must be assessed in order to prepare further tests. For example, this includes:
    - Cryptographic algorithms in use
    - Authentication and authorization mechanism
    - Local paths and environment details

Remediation

As mentioned above, the disclosure of such information is not considered a vulnerability. However, in order to impede exploitation attempts, only information necessary for the device operation should be stored on it.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta

This test case is based on: ISTG-FW-INFO-002.

Disclosure of Ecosystem Details (ISTG-MEM-INFO-003)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1*

Summary

The contents of the device memory might disclose information about the surrounding IoT ecosystem, e.g., sensitive URLs, IP addresses, software in use etc. An attacker might be able to use this information to prepare and execute attacks against the ecosystem.

For example, relevant information might be included in files of various types like configuration files and text files.

Test Objectives

- It must be determined if the data stored in the device memory, e.g., configuration files, contain relevant information about the surrounding ecosystem.

Remediation

The disclosure of information should be reduced to the minimum, which is required for operating the device. The disclosed information has to be assessed and all unnecessarily included data should be removed.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta

This test case is based on: ISTG-FW-INFO-003.

Disclosure of User Data (ISTG-MEM-INFO-004)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1*

Summary

During runtime, a device is accumulating and processing data of different kinds, such as personal data of its users. If this data is not stored securely, an attacker might be able to recover it from the device.

Test Objectives

- It has to be checked whether user data can be accessed by unauthorized individuals.

Remediation

Access to user data should only be granted to individuals and processes that need to have access to it. No unauthorized or not properly authorized individual should be able to recover user data.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta

This test case is based on: ISTG-FW[INST]-INFO-001.

Secrets (ISTG-MEM-SCRT)

IoT devices are often operated outside of the control space their manufacturer. Still, they need to establish connections to other network nodes within the IoT ecosystem, e.g., to request and receive firmware updates or to send data to a cloud API. Hence, it might be required that the device can provide some kind of authentication credential or secret. These secrets need to be stored on the device in a secure manner to prevent them from being stolen and used to impersonate the device.

Unencrypted Storage of Secrets (ISTG-MEM-SCRT-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1*

Summary

Sensitive data and secrets should be stored in an encrypted manner, so that even if an attacker has managed to get access to the memory, he has no access to the respective plaintext data.

The strength of the cryptographic algorithms in use will be covered by ISTG-MEM-CRYPT-001 and has no relevance for this test case.

Test Objectives

- By searching the contents of the device memory, it must be determined whether it includes secrets in plaintext form.

Remediation

Secrets have to be stored using proper cryptographic algorithms. Only the encrypted form of the secret should be stored.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta

This test case is based on: ISTG-FW-SCRT-002.

Cryptography (ISTG-MEM-CRYPT)

Many IoT devices need to implement cryptographic algorithms, e.g., to securely store sensitive data, for authentication purposes or to receive and verify encrypted data from other network nodes. Failing to implement secure, state of the art cryptography might lead to the exposure of sensitive data, device malfunctions or loss of control over the device.

Usage of Weak Cryptographic Algorithms (ISTG-MEM-CRYPT-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1*

Summary

Cryptography can be implemented in various ways. However, due to evolving technologies, new algorithms and more computing power becoming available, many old cryptographic algorithms are nowadays considered weak or insecure. Thus, either new and stronger cryptographic algorithms have to be used or existing algorithms must be adapted, e.g., by increasing the key length or using alternative modes of operation.

The usage of weak cryptographic algorithms might allow an attacker to recover the plaintext from a given ciphertext in a timely manner.

Test Objectives

- The data, stored on the device, must be checked for the presence of encrypted data segments. In case that encrypted data segments are found, it must be checked whether the cryptographic algorithms in use can be identified.
- Furthermore, based on ISTG-MEM-INFO-001 and ISTG-MEM-INFO-002, it must be checked whether any source code, configuration files etc. disclose the usage of certain cryptographic algorithms.
- In case that cryptographic algorithms can be identified, it must be determined whether the algorithms in use and their configuration are providing a sufficient level of security at the time of testing, e.g., by consulting cryptography guidelines like the technical guideline TR-02102-1 by the BSI.

Remediation

Only strong, state of the art cryptographic algorithms should be used. Furthermore, these algorithms must be used in a secure manner by setting proper parameters, such as an appropriate key length or mode ofoperation.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta

This test case is based on: ISTG-FW-CRYPT-001.

3.3. Firmware (ISTG-FW)

Table of Contents

Overview

This section includes test cases and categories for the component firmware and the component specializations installed firmware (ISTG-FW[INST]) and firmware update mechanism (ISTG-FW[UPDT]) respectively. The firmware might be accessible with all physical access levels, depending on how this access is implemented in detail.

In regards to test case categories that are relevant for processing units, the following were identified:

- **Information Gathering:** Focuses on information that is stored within the firmware and that might be disclosed to potential attackers if not being properly protected or removed.

- **Configuration and Patch Management:** Focuses on vulnerabilities and issues in the configuration of a firmware and its software components.

- **Secrets:** Focuses on secrets that are stored within the firmware in an insecure manner.

- **Cryptography:** Focuses on vulnerabilities in the cryptographic implementation.

- **Authorization (Installed Firmware):** Focuses on vulnerabilities that allow to get unauthorized access to the firmware or to elevate privileges in order to access restricted functionalities.
- **Business Logic (Firmware Update Process):** Focuses on vulnerabilities in the design and implementation of the firmware update process.

All test cases and categories for the component ISTG-FW focus on generic firmware analysis aspects, without regards to the specifics of specializations for this component.

Information Gathering (ISTG-FW-INFO)

The firmware of an IoT device can include various information, which, if disclosed, could reveal details regarding the inner workings of the device or the surrounding IoT ecosystem to potential attackers. This could enable and facilitate further, more advanced attacks.

Disclosure of Source Code and Binaries (ISTG-FW-INFO-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on which component specialization should be tested and how i

Summary

The disclosure of uncompiled source code could accelerate the exploitation of the software implementation since vulnerabilities can be directly identified in the code without the need to perform tests in a trial and error manner. Furthermore, left-over source code might include internal development information, developer comments or hard-coded sensitive data, which were not intended for productive use.

Similar to uncompiled source code, compiled binaries might also disclose relevant information. However, reverse-engineering might be required to retrieve useful data, which could take a considerable amount of time. Thus, the tester has to assess which binaries might be worth analyzing, ideally in coordination with the firmware manufacturer.

Test Objectives

- It must be checked if uncompiled source code can be identified within the firmware.
- If uncompiled source code is detected, its content must be analyzed for the presence of sensitive data, which might be useful for potential attackers (also see ISTG-FW-SCRT-

003).

- Reverse-engineering of selected binaries should be performed in order to obtain useful information regarding the firmware implementation and the processing of sensitive data.

Remediation

If possible, uncompiled source code should be removed from firmware, intended for productive use. If the source code has to be included, it must be verified that all internal development data is removed before the firmware is released.

Since it is not possible to prevent reverse-engineering completely, measures to restrict access to the firmware in general should be implemented to reduce the attack surface. Furthermore, the reverse-engineering process can be impeded, e.g., by obfuscating the code.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Disclosure of Implementation Details (ISTG-FW-INFO-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If details about the implementation, e.g., algorithms in use or the authentication procedure, are available to potential attackers, flaws and entry points for successful attacks are easier to detect. While the disclosure of such details alone is not considered to be a vulnerability, it facilitates the identification of potential attack vectors, thus allowing an attacker to exploit insecure implementations faster.

For example, relevant information might be included in files of various types like configuration files, text files, system settings or databases.

Test Objectives

- Accessible details regarding the implementation must be assessed in order to prepare further tests. For example, this includes:
  - Cryptographic algorithms in use
  - Authentication and authorization mechanisms
  - Local paths and environment details

Remediation

As mentioned above, the disclosure of such information is not considered a vulnerability. However, in order to impede exploitation attempts, only information, necessary for the device operation, should be accessible.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Disclosure of Ecosystem Details (ISTG-FW-INFO-003)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

The contents of the device firmware might disclose information about the surrounding IoT ecosystem, e.g., sensitive URLs, IP addresses, software in use etc. An attacker might be able to use this information to prepare and execute attacks against the ecosystem.

For example, relevant information might be included in files of various types like configuration files and text files.

Test Objectives

- It must be determined if (parts of) the firmware, e.g., configuration files, contain relevant information about the surrounding ecosystem.

Remediation

The disclosure of information should be reduced to the minimum, which is required for operating the device. The disclosed information has to be assessed and all unnecessarily included data should be removed.

References

For this test case, data from the following available sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Configuration and Patch Management (ISTG-FW-CONF)

Since IoT devices can have a long lifespan, it is important to make sure that the software, running on the device, is regularly updated in order to apply the latest security patches. The update process of the firmware itself will be covered by ISTG-FW[UPDT]. However, it must also be verified that software packages, which are included in the firmware, are up-to-date as well.

Usage of Outdated Software (ISTG-FW-CONF-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software is potentially vulnerable to attacks. For example, coding errors could lead to undefined program behavior, which then can be exploited by an attacker to gain access to data, processed by the application, or to perform actions in the context of the runtime environment. Furthermore, vulnerabilities in the used frameworks, libraries and other technologies might also affect the security level of a given piece of software.

Usually, developers release an update once a vulnerability was detected in their software. These updates should be installed as soon as possible in order to reduce the probability of successful attacks. Otherwise, attackers could use known vulnerabilities to perform attacks against the device.

Test Objectives

- The version identifiers of installed software packages as well as libraries and frameworks in use must be determined.

- Based on the detected version identifiers, it must be determined if the software version in use is up-to-date, e.g., by consulting the website of the software developer or public repositories.

- By using vulnerability databases, such as the National Vulnerability Database of the NIST, it has to be checked whether any vulnerabilities are known for the detected software versions.

Remediation

The firmware should not include any outdated software packages. A proper patch management process, which ensures that applicable updates are installed once being available, should be implemented.

References

For this test case, data from the following available sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Presence of Unnecessary Software and Functionalities (ISTG-FW-CONF-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys

Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software, which is included in the firmware, broadens the attack surface since it might be used to perform attacks against the device. Even if the installed software is up-to-date, it might still be affected by unpublished vulnerabilities. It is also possible that a software program facilitates an attack without being vulnerable, e.g., by providing access to specific files or processes.

Test Objectives

- A list of software packages, that are included in the firmware, should be assembled.
- Based on the device documentation, its behavior and the intended use cases, it must be determined whether any of the installed software packages are not mandatory for the device operation.

Remediation

The attack surface should be minimized as much as possible by removing or disabling every software that is not required for the device operation.

Especially in case of general-purpose operating systems, such as Windows and Linux systems, it must be ensured that any unnecessary operating system features are disabled.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Secrets (ISTG-FW-SCRT)

IoT devices are often operated outside of the control space of their manufacturer. Still, they need to establish connections to other network nodes within the IoT ecosystem, e.g., to request and receive firmware updates or to send data to a cloud API. Hence, it might be required that the device has to provide some kind of authentication credential or secret. These secrets need to be stored on the device in a secure manner to prevent them from being stolen and used to impersonate the device.

Secrets Stored in Public Storage (ISTG-FW-SCRT-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Generally, there are multiple kinds of storage spaces within a file system, some of which are publicly available and some that can only be accessed with a certain level of privileges. If sensitive data or secrets are stored in publicly accessible storage spaces, users who should not have access to this data but who have access to the file system could read or modify it. In case of a successful attack, it is very likely that secrets, stored in public storage, are disclosed.

Test Objectives

- Files and databases within public storage spaces must be checked for the presence of secrets, such as passwords, symmetric or private keys and tokens.

Remediation

Access to secrets should only be granted to the accounts or processes with proper privileges. Thus, secrets should be stored in protected storage areas or designated key stores that are only available to certain entities.

References

For this test case, data from the following available sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Unencrypted Storage of Secrets (ISTG-FW-SCRT-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys

Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Sensitive data and secrets should be stored in an encrypted manner, so that even if an attacker has managed to get access to it, he has no access to the respective plaintext data.

Contrary to ISTG-FW-SCRT-001, it does not matter if the secrets are stored in public or restricted storage spaces, since it is assumed that the attacker has already gotten access to the data, e.g., by circumventing access restrictions or by exploiting a process with access to the restricted storage. Furthermore, the strength of the cryptographic algorithms in use will be covered by ISTG-FW-CRYPT-001 and has no relevance for this test case.

Test Objectives

- By searching public and restricted storage spaces, it must be determined whether the firmware includes secrets in plaintext form.

Remediation

Secrets have to be stored using proper cryptographic algorithms. Only the encrypted form of the secret should be stored.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Usage of Hardcoded Secrets (ISTG-FW-SCRT-003)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Sometimes, developers tend to incorporate secrets directly into the source code of their software. This can lead to a variety of security issues like:

- the disclosure of secrets via published source code snippets or decompiled source code,
- endangering all devices that are using the given software since it is very likely that the same secret is used on all devices (otherwise, the source code needs to be changed and compiled for every device individually) and
- impeding reactive measures in case of the secret being compromised since changing the secret requires a software update.

Test Objectives

- Based on ISTG-FW-INFO-001, it must be checked if any hard-coded secrets can be identified.

Remediation

Secrets should not be hard-coded into the source code. Instead, secrets should be stored in a secure manner (see ISTG-FW-SCRT-001 and ISTG-FW-SCRT-002) and the software process should dynamically retrieve the secrets from the secure storage during runtime.

References

For this test case, data from the following available sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Cryptography (ISTG-FW-CRYPT)

Many IoT devices need to implement cryptographic algorithms, e.g., to securely store sensitive data, for authentication purposes or to receive and verify encrypted data from other network nodes. Failing to implement secure, state of the art cryptography might lead to the exposure of sensitive data, device malfunctions or loss of control over the device.

Usage of Weak Cryptographic Algorithms (ISTG-FW-CRYPT-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Cryptography can be implemented in various ways. However, due to evolving technologies, new algorithms and more computing power becoming available, many old cryptographic algorithms are nowadays considered weak or insecure. Thus, either new and stronger cryptographic algorithms have to be used or existing algorithms must be adapted, e.g., by increasing the key length or using alternative modes of operation.

The usage of weak cryptographic algorithms might allow an attacker to recover the plaintext from a given ciphertext in a timely manner.

Test Objectives

- The data, stored by or within the firmware, must be checked for the presence of encrypted data segments. In case that encrypted data segments are found, it must be checked whether the cryptographic algorithms in use can be identified.

- Furthermore, based on ISTG-FW-INFO-001 and ISTG-FW-INFO-002, it must be checked whether any source code, configuration files etc. disclose the usage of certain cryptographic algorithms.

- In case that cryptographic algorithms can be identified, it must be determined whether the algorithms in use and their configuration are providing a sufficient level of security at the time of testing, e.g., by consulting cryptography guidelines like the technical guideline TR-02102-1 by the BSI.

Remediation

Only strong, state of the art cryptographic algorithms should be used. Furthermore, these algorithms must be used in a secure manner by setting proper parameters, such as an appropriate key length or mode of operation.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta

- "The IoT Hacker's Handbook" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

3.3.1. Installed Firmware (ISTG-FW[INST])

Table of Contents

Overview

One specialization of the component firmware is the installed form of a firmware, which might be the subject of a dynamic analysis. The dynamic analysis allows to test the handling of data during runtime. This way, the processing and storing of user data can also be analyzed. As a pre-requisite for the dynamic analysis, a device, which is running the target firmware version, must be provided.

Authorization (ISTG-FW[INST]-AUTHZ)

Usually, only certain individuals, e.g., administrators, should be allowed to access the device firmware during runtime. Thus, proper authentication and authorization procedures need to be in place, which ensure that only authorized users can get access to the firmware.

Unauthorized Access to the Firmware (ISTG-FW[INST]-AUTHZ-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1*

Summary

Depending on the specific implementation of a given device, access to the firmware or its functions might be restricted to individuals with a certain authorization access level, e.g., *AA-2*, *AA-3* or *AA-4*. If the device firmware fails to correctly verify access permissions, any attacker (*AA-1*) might be able to get access to the firmware.

Test Objectives

- It must be checked if authorization checks for access to the firmware are implemented.
- In case that authorization checks are in place, it must be determined whether there is a way to bypass them.

Remediation

Proper authorization checks need to be implemented, which ensure that access to the firmware is only possible for authorized individuals.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-AUTHZ-001.

Privilege Escalation (ISTG-FW[INST]-AUTHZ-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-3* (depending on the access model for the given device)

Summary

Depending on the specific implementation of a given device, access to parts of the firmware or its functions might be restricted to individuals with a certain authorization access level, e.g., *AA-3* or *AA-4*. If the device firmware fails to correctly verify access permissions, an attacker with a lower authorization access level than intended might be able to get access to the restricted firmware parts.

Test Objectives

- Based on *ISTG-FW-AUTHZ-001*, it must be determined whether there is a way to elevate the given access privileges and thus to access restricted functions or parts of the firmware.

Remediation

Proper authorization checks need to be implemented, which ensure that access to restricted parts of the firmware is only possible for individuals with the required authorization access levels.

References

This test case is based on: ISTG-DES-AUTHZ-002.

Information Gathering (ISTG-FW[INST]-INFO)

As mentioned above, during the dynamic analysis, it is also possible to test whether user data is securely stored on the device during runtime.

Disclosure of User Data (ISTG-FW[INST]-INFO-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

During runtime, a device is accumulating and processing data of different kinds, such as personal data of its users. If this data is not stored securely, an attacker might be able to recover it from the device.

Test Objectives

- It has to be checked whether user data can be accessed by unauthorized individuals.

Remediation

Access to user data should only be granted to individuals and processes that need to have access to it. No unauthorized or not properly authorized individual should be able to recover user data.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Cryptography (ISTG-FW[INST]-CRYPT)

Many IoT devices need to implement cryptographic algorithms, e.g., to securely store sensitive data, for authentication purposes or to receive and verify encrypted data from other network nodes. Failing to implement secure, state of the art cryptography might lead to the exposure of sensitive data, device malfunctions or loss of control over the device.

Insufficient Verification of the Bootloader Signature (ISTG-FW[INST]-CRYPT-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Verifying the digital signature of the bootloader is an important measure to detect manipulations of the bootloader, thus preventing the execution of manipulated firmware on a device.

Test Objectives

- It must be checked if the signature of the bootloader is properly verified by the device during the boot process.

Remediation

The device must properly verify the digital signature of a bootloader before it is executed. A bootloader without a valid signature should not be executed.

3.3.2. Firmware Update Mechanism (ISTG-FW[UPDT])

Table of Contents

Overview

Another important aspect of the device firmware is the firmware update mechanism. Failing to implement a secure update mechanism might enable attackers to install a custom, manipulated firmware on the device, thus gaining complete control over it.

The following categories are not inherited by the specialization ISTG-FW[UPDT]:

- **Configuration and Patch Management (ISTG-FW-CONF)**: This category focuses on the configuration and patch management aspects of a firmware file. Since ISTG-FW[UPDT] focuses on the firmware update mechanism rather than a specific firmware file, the respective test cases are not applicable.

- **Secrets (ISTG-FW-SCRT)**: This category focuses on the handling of secrets within a firmware file. Since ISTG-FW[UPDT] focuses on the firmware update mechanism rather than a specific firmware file, the respective test cases are not applicable.

Authorization (ISTG-FW[UPDT]-AUTHZ)

Since the test of the firmware update mechanism is also a dynamic analysis, it is possible to check if only authorized individuals can initialize and perform an update.

Unauthorized Firmware Update (ISTG-FW[UPDT]-AUTHZ-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-3* (depending on the access model for the given device)

Summary

Depending on the specific implementation of a given device, the permission to perform firmware updates might be restricted to individuals with a certain authorization access level, e.g., *AA-2*, *AA-3* or *AA-4*. If the device firmware fails to correctly verify these permissions, any attacker (*AA-1*) or an attacker with a lower authorization access level than intended might be able to perform unintended firmware updates.

Test Objectives

- It must be checked if authorization checks for performing a firmware update are implemented.
- In case that authorization checks are in place, it must be determined whether there is a way to bypass them.

Remediation

Proper authorization checks need to be implemented, which ensure that a firmware update can only be performed by individuals with certain authorization access levels.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Cryptography (ISTG-FW[UPDT]-CRYPT)

During the firmware update process, cryptographic algorithms are used to verify the integrity of the new firmware and to ensure that no sensitive data is disclosed in transit.

Insufficient Firmware Update Signature (ISTG-FW[UPDT]-CRYPT-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

One way to manipulate a device would be to install a manipulated firmware package. In order to enable the detection of modifications, the firmware update package needs to be digitally signed. This way, the validity of the package can be verified during the installation or update process.

Test Objectives

- It must be determined if a digital signature for the firmware update package is available.

- If a digital signature is available, it must be checked whether the validity of the signature can be verified.

- Based on ISTG-FW-CRYPT-001, the cryptographic algorithm, used for generating the digital signature, has to be assessed in order to determine whether a weak our outdated algorithm was used.

Remediation

A valid digital signature must be available for the firmware update package. Furthermore, it must be possible to verify the validity of the digital signature.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Insufficient Firmware Update Encryption (ISTG-FW[UPDT]-CRYPT-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Firmware update packages might include confidential data of the soft- and hardware developer, e.g., intellectual property. Hence, it might be required to encrypt the package itself.

Test Objectives

- It has to be clarified with the firmware developer whether the firmware update package needs to be encrypted.
- If encryption is required, it must be determined whether the package is encrypted.
- Based on ISTG-FW-CRYPT-001, it has to be determined whether proper algorithms were used for encryption.

Remediation

The firmware update package should be encrypted using state of the art cryptographic algorithms.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Insecure Transmission of the Firmware Update (ISTG-FW[UPDT]-CRYPT-003)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If the firmware update process is not performed via a secure channel or if no further measures are in place to ensure the confidentiality and to detect modifications of the transmitted data, an attacker with access to the communication channel might be able to interfere with the update process.

Test Objectives

- It has to be determined whether the firmware update is performed over a secure channel.
- If the firmware update is performed over an insecure channel, like the internet, it must be checked whether proper measures in regards of confidentiality and integrity are in place.
- If, for example, the communication channel is secured using TLS, it must be checked which cipher suites are supported and if the server certificate is validated by the client.

Remediation

If feasible, the firmware update should be performed via a secure channel. Otherwise, proper measures need to be implemented in order to prevent or detect interferences with potential attackers.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Insufficient Verification of the Firmware Update Signature (ISTG-FW[UPDT]-CRYPT-004)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Even if the firmware update package is digitally signed, an attacker could install a manipulated firmware package on the device in case that the digital signature is not properly validated. For example, the device might not reject the update if no signature is provided.

Test Objectives

- Based on ISTG-FW-CRYPT-001, it must be checked if the signature of the firmware update package is properly verified by the device during the update process.

Remediation

The device must properly verify the digital signature of an update package before the installation process is started. Any update package without a valid signature or with no signature at all should be rejected.

References

For this test case, data from the following sources was consolidated:

- OWASP "Firmware Security Testing Methodology"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Business Logic (ISTG-FW[UPDT]-LOGIC)

Even if all other aspects of the firmware update are securely implemented, issues in the underlying logic of the update process itself might render the device vulnerable to attacks. Thus, it must be verified if the process is working as intended and if exceptions are detected and properly handled.

Insufficient Rollback Protection (ISTG-FW[UPDT]-LOGIC-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the firmware can be accessed, e.g., via an internal/phys
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Some manufacturers implement a rollback protection for their devices. This rollback protection prevents updating a device firmware to an older version than the currently installed one. This way, an attacker can not install a valid but outdated firmware in order to exploit known vulnerabilities of that version.

Test Objectives

- It has to be assessed whether it is possible to install older versions of the firmware.

Remediation

A proper rollback protection mechanism verifying that the firmware version to be installed is newer than the currently installed version should be implemented.

References

For this test case, data from the following sources was consolidated:

- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

3.4. Data Exchange Services (ISTG-DES)

Table of Contents

Overview

This section includes test cases and categories for the component data exchange service. Based on its implementation and intended use, a data exchange service might be accessible with all physical access levels.

In regards to test case categories that are relevant for data exchange service, the following were identified:

- **Authorization:** Focuses on vulnerabilities that allow to get unauthorized access to the data exchange process or to elevate privileges in order to access restricted functionalities.

- **Information Gathering:** Focuses on information that is handled by the data exchange service and that might be disclosed to potential attackers if not being properly protected or removed.
- **Configuration and Patch Management:** Focuses on vulnerabilities and issues in the configuration of a data exchange service and its software components.
- **Secrets:** Focuses on secrets that are handled by the data exchange service in an insecure manner.
- **Cryptography:** Focuses on vulnerabilities in the cryptographic implementation.
- **Business Logic:** Focuses on vulnerabilities in the implementation of the data exchange service.
- **Input Validation:** Focuses on vulnerabilities regarding the validation and processing of input from untrustworthy sources.

Authorization (ISTG-DES-AUTHZ)

Depending on the access model for a given device, only certain individuals might be allowed to access a data exchange service. Thus, proper authentication and authorization procedures need to be in place, which ensure that only authorized users can get access.

Unauthorized Access to the Data Exchange Service (ISTG-DES-AUTHZ-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1*

Summary

Depending on the specific implementation of a given device, access to a data exchange service might be restricted to individuals with a certain authorization access level, e.g., *AA-2*, *AA-3* or *AA-4*. If the device fails to correctly verify access permissions, any attacker (*AA-1*) might be able to get access.

Test Objectives

- It must be checked if authorization checks for access to the data exchange service are implemented.
- In case that authorization checks are in place, it must be determined whether there is a way to bypass them.

Remediation

Proper authorization checks need to be implemented, which ensure that access to the data exchange service is only possible for authorized individuals.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Privilege Escalation (ISTG-DES-AUTHZ-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-2 - AA-3* (depending on the access model for the given device)

Summary

Depending on the specific implementation of a given device, access to some functionalities via a data exchange service might be restricted to individuals with a certain authorization access level, e.g., *AA-3* or *AA-4*. If the device fails to correctly verify access permissions, an attacker with a lower authorization access level than intended might be able to get access to the restricted functionalities.

Test Objectives

- Based on ISTG-DES-AUTHZ-001, it must be determined whether there is a way to elevate the given access privileges and thus to access restricted functionalities.

Remediation

Proper authorization checks need to be implemented, which ensure that access to restricted functionalities is only possible for individuals with the required authorization access levels.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"

- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Information Gathering (ISTG-DES-INFO)

Data exchange service might disclose various information, which could reveal details regarding the inner workings of the device or the surrounding IoT ecosystem to potential attackers. This could enable and facilitate further, more advanced attacks.

Disclosure of Implementation Details (ISTG-DES-INFO-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If details about the implementation, e.g., algorithms in use or the authentication procedure, are available to potential attackers, flaws and entry points for successful attacks are easier to detect. While the disclosure of such details alone is not considered to be a vulnerability, it facilitates the identification of potential attack vectors, thus allowing an attacker to exploit insecure implementations faster.

For example, relevant information might be included in service banners, response headers or error messages.

Test Objectives

- Accessible details regarding the implementation must be assessed in order to prepare further tests. For example, this includes:
  - Cryptographic algorithms in use
  - Authentication and authorization mechanisms
  - Local paths and environment details

Remediation

As mentioned above, the disclosure of such information is not considered a vulnerability. However, in order to impede exploitation attempts, only information, necessary for the device operation, should be displayed.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-001.

Disclosure of Ecosystem Details (ISTG-DES-INFO-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

A data exchange service might disclose information about the surrounding IoT ecosystem, e.g., sensitive URLs, IP addresses, software in use etc. An attacker might be able to use this information to prepare and execute attacks against the ecosystem.

For example, relevant information might be included in service banners, response headers or error messages.

Test Objectives

- It must be determined if the data exchange service discloses relevant information about the surrounding ecosystem.

Remediation

The disclosure of information should be reduced to the minimum, which is required for operating the device. The disclosed information it has to be assessed and all unnecessarily included data should be removed.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"

- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-003.

Disclosure of User Data (ISTG-DES-INFO-003)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

During runtime, a device is accumulating and processing data of different kinds, such as personal data of its users. If this data is disclosed, an attacker might be able to get access to it.

Test Objectives

- It has to be checked whether user data can be accessed by unauthorized individuals.

Remediation

Access to user data should only be granted to individuals and processes that need to have access to it. No unauthorized or not properly authorized individual should be able to access user data.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW[INST]-INFO-001.

Configuration and Patch Management (ISTG-DES-CONF)

Since IoT devices can have a long lifespan, it is important to make sure that the software, running on the device, is regularly updated in order to apply the latest security patches. The update process of the firmware itself will be covered by ISTG-FW[UPDT]. However, it must also be verified that software packages, which are running on the device and are handling data exchange processes, are up-to-date as well.

Usage of Outdated Software (ISTG-DES-CONF-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software is potentially vulnerable to attacks. For example, coding errors could lead to undefined program behavior, which then can be exploited by an attacker to gain access to data, processed by the application, or to perform actions in the context of the runtime environment. Furthermore, vulnerabilities in the used frameworks, libraries and other technologies might also affect the security level of a given piece of software.

Usually, developers release an update once a vulnerability was detected in their software. These updates should be installed as soon as possible in order to reduce the probability of successful attacks. Otherwise, attackers could use known vulnerabilities to perform attacks against the device.

Test Objectives

- The version identifiers of installed software packages as well as libraries and frameworks in use must be determined.
- Based on the detected version identifiers, it must be determined if the software version in use is up-to-date, e.g., by consulting the website of the software developer or public repositories.
- By using vulnerability databases, such as the National Vulnerability Database of the NIST, it has to be checked whether any vulnerabilities are known for the detected software versions.

Remediation

No outdated software packages should be running on the device. A proper patch management process, which ensures that applicable updates are installed once being available, should be implemented.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-001.

Presence of Unnecessary Software and Functionalities (ISTG-DES-CONF-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software, which is available on the device, broadens the attack surface since it might be used to perform attacks against the device. Even if the installed software is up-to-date, it might still be affected by unpublished vulnerabilities. It is also possible that a software program facilitates an attack without being vulnerable, e.g., by providing access to specific files or processes.

Test Objectives

- A list of functionalities, available via the data exchange process, should be assembled.

- Based on the device documentation, its behavior and the intended use cases, it must be determined whether any of the available functionalities are not mandatory for the device operation.

Remediation

The attack surface should be minimized as much as possible by removing or disabling every software that is not required for the device operation.

Especially in case of general-purpose operating systems, such as Windows and Linux systems, it must be ensured that any unnecessary operating system features are disabled.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-002.

Secrets (ISTG-DES-SCRT)

IoT devices are often operated outside of the control space of their manufacturer. Still, they need to establish connections to other network nodes withinthe IoT ecosystem, e.g., to request and receive firmware updates or to send data to a cloud API. Hence, it might be required that the device has to provide some kind of authentication credential or secret. These secrets need to be stored on the device in a secure manner to prevent them from being stolen and used to impersonate the device.

Access to Confidential Data (ISTG-DES-SCRT-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Malfunctions, unintended behavior or improper implementation of a data exchange service might enable an attacker to get access to secrets.

Test Objectives

- It has to be determined whether secrets can be accessed via the data exchange service.

Remediation

Access to secrets should only be granted to individuals and processes that need to have access to them. No unauthorized or not properly authorized individual should be able to access secrets.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Cryptography (ISTG-DES-CRYPT)

Many IoT devices need to implement cryptographic algorithms, e.g., to securely store sensitive data, for authentication purposes or to receive and verify encrypted data from other network nodes. Failing to implement secure, state of the art cryptography might lead to the exposure of sensitive data, device malfunctions or loss of control over the device.

Usage of Weak Cryptographic Algorithms (ISTG-DES-CRYPT-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Cryptography can be implemented in various ways. However, due to evolving technologies, new algorithms and more computing power becoming available, many old cryptographic algorithms are nowadays considered weak or insecure. Thus, either new and stronger cryptographic algorithms have to be used or existing algorithms must be adapted, e.g., by increasing the key length or using alternative modes of operation.

The usage of weak cryptographic algorithms might allow an attacker to recover the plaintext from a given ciphertext in a timely manner.

Test Objectives

- The data, processed by the data exchange service, must be checked for the presence of encrypted data segments. In case that encrypted data segments are found, it must be checked whether the cryptographic algorithms in use can be identified.

- Furthermore, based on ISTG-DES-INFO-001, it must be checked whether headers, system messages etc. disclose the usage of certain cryptographic algorithms.

- In case that cryptographic algorithms can be identified, it must be determined whether the algorithms in use and their configuration are providing a sufficient level of security at the time of testing, e.g., by consulting cryptography guidelines like the technical guideline TR-02102-1 by the BSI.

Remediation

Only strong, state of the art cryptographic algorithms should be used. Furthermore, these algorithms must be used in a secure manner by setting proper parameters, such as an appropriate key length or mode of operation.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CRYPT-001.

Business Logic (ISTG-DES-LOGIC)

Even if all other aspects of the data exchange service are securely implemented and configured, issues in the underlying logic itself might render the device vulnerable to attacks. Thus, it must be verified if the data exchange service and its functionalities are working as intended and if exceptions are detected and properly handled.

Circumvention of the Intended Business Logic (ISTG-DES-LOGIC-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Flaws in the implementation of the business logic might result in unintended behavior or malfunctions of the device. For example, if an attacker intentionally misses to provide relevant input data or tries to skip or change important steps in the processing workflow the device might end up in an unknown, potentially insecure state.

Test Objectives

- Based on the specific business logic implementation, it has to be determined whether deviations from the defined workflows are properly detected and handled.

Remediation

The device should not end up in an unknown state. Anomalies in the workflow must be detected and exceptions have to be handled properly.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Input Validation (ISTG-DES-INPV)

In order to ensure that only valid and well-formed data enters the processing flows of a device, the input from a all untrustworthy sources, e.g., users or external systems, has to be verified and validated.

Insufficient Input Validation (ISTG-DES-INPV-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place, an attacker might be able to submit arbitrary and malformed data. Thus, the process, which handles the user input, or another downstream component might stop working properly due to not being able to process the data. This could result in malfunctions that might enable an attacker to manipulate the device behavior or render it unavailable.

Test Objectives

- It must be determined whether input to the data exchange service is validated.
- In case that an input validation mechanism is implemented, it has to be checked if there is a way to submit data, which does not comply with the intended data structure and value ranges.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

Code or Command Injection (ISTG-DES-INPV-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending on how the data exchange service can be accessed, e.g., if it was designed for remote access)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place an attacker might be able to submit code or commands, which then might be executed by the system. It strictly depends on the specific implementation of the device and the data exchange service which code and commands are potentially executable. For example, possible injection attacks are Cross Site Scripting, SQL injection and OS command injection.

Test Objectives

- Based on ISTG-DES-INPV-001, it must be checked whether it is possible to submit code or commands, which are then executed by the system.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

3.5. Internal Interfaces (ISTG-INT)

Table of Contents

Overview

This section includes test cases and categories for the component internal interface. Similar to the processing unit and the memory, the internal interface is a device-internal element that can only be accessed with *PA-4*. Establishing a direct connection to an internal interface might require specific hardware equipment (e.g., a debugging board, an oscilloscope or test probes).

In regards to test case categories that are relevant for an internal interface, the following were identified:

- **Authorization:** Focuses on vulnerabilities that allow to get unauthorized access to the internal interface or to elevate privileges in order to access restricted functionalities.

- **Information Gathering:** Focuses on information that is handled by the internal interface and that might be disclosed to potential attackers if not being properly protected or removed.
- **Configuration and Patch Management:** Focuses on vulnerabilities and issues in the configuration of an internal interface and its software components.
- **Secrets:** Focuses on secrets that are handled by the internal interface in an insecure manner.
- **Cryptography:** Focuses on vulnerabilities in the cryptographic implementation.
- **Business Logic:** Focuses on vulnerabilities in the implementation of the internal interface.
- **Input Validation:** Focuses on vulnerabilities regarding the validation and processing of input from untrustworthy sources.

Authorization (ISTG-INT-AUTHZ)

Depending on the access model for a given device, only certain individuals might be allowed to access an internal interface. Thus, proper authentication and authorization procedures need to be in place, which ensure that only authorized users can get access.

Unauthorized Access to the Interface (ISTG-INT-AUTHZ-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1*

Summary

Depending on the specific implementation of a given device, access to an internal interface might be restricted to individuals with a certain authorization access level, e.g., *AA-2*, *AA-3* or *AA-4*. If the device fails to correctly verify access permissions, any attacker (*AA-1*) might be able to get access.

Test Objectives

- It must be checked if authorization checks for access to the internal interface are implemented.
- In case that authorization checks are in place, it must be determined whether there is a way to bypass them.

Remediation

Proper authorization checks need to be implemented, which ensure that access to the internal interface is only possible for authorized individuals.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-AUTHZ-001.

Privilege Escalation (ISTG-INT-AUTHZ-002)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-2 - AA-3* (depending on the access model for the given device)

Summary

Depending on the specific implementation of a given device, access to some functionalities via an internal interface might be restricted to individuals with a certain authorization access level, e.g., *AA-3* or *AA-4*. If the interface fails to correctly verify access permissions, an attacker with a lower authorization access level than intended might be able to get access to the restricted functionalities.

Test Objectives

- Based on ISTG-INT-AUTHZ-001, it must be determined whether there is a way to elevate the given access privileges and thus to access restricted functionalities.

Remediation

Proper authorization checks need to be implemented, which ensure that access to restricted functionalities is only possible for individuals with the required authorization access levels.

References

This test case is based on: ISTG-DES-AUTHZ-002.

Information Gathering (ISTG-INT-INFO)

Internal interfaces might disclose various information, which could reveal details regarding the inner workings of the device or the surrounding IoT ecosystem to potential attackers. This could enable and facilitate further, more advanced attacks.

Disclosure of Implementation Details (ISTG-INT-INFO-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If details about the implementation, e.g., algorithms in use or the authentication procedure, are available to potential attackers, flaws and entry points for successful attacks are easier to detect. While the disclosure of such details alone is not considered to be a vulnerability, it facilitates the identification of potential attack vectors, thus allowing an attacker to exploit insecure implementations faster.

For example, relevant information might be included in service banners, console output or error messages.

Test Objectives

- Accessible details regarding the implementation must be assessed in order to prepare further tests. For example, this includes:
  - Cryptographic algorithms in use
  - Authentication and authorization mechanisms
  - Local paths and environment details

Remediation

As mentioned above, the disclosure of such information is not considered a vulnerability. However, in order to impede exploitation attempts, only information, necessary for the device operation, should be displayed.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta

- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-002.

Disclosure of Ecosystem Details (ISTG-INT-INFO-002)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* depending on the access model for the given device)

Summary

An internal interface might disclose information about the surrounding IoT ecosystem, e.g., sensitive URLs, IP addresses, software in use etc. An attacker might be able to use this information to prepare and execute attacks against the ecosystem.

For example, relevant information might be included in service banners, console output or error messages.

Test Objectives

- It must be determined if the internal interface discloses relevant information about the surrounding ecosystem.

Remediation

The disclosure of information should be reduced to the minimum, which is required for operating the device. The disclosed information it has tobe assessed and all unnecessarily included data should be removed.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-003.

Disclosure of User Data (ISTG-INT-INFO-003)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

During runtime, a device is accumulating and processing data of different kinds, such as personal data of its users. If this data is disclosed, an attacker might be able to get access to it.

Test Objectives

- It has to be checked whether user data can be accessed by unauthorized individuals.

Remediation

Access to user data should only be granted to individuals and processes that need to have access to it. No unauthorized or not properly authorized individual should be able to access user data.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW[INST]-INFO-001.

Configuration and Patch Management (ISTG-INT-CONF)

Since IoT devices can have a long lifespan, it is important to make sure that the software, running on the device, is regularly updated in order to apply the latest security patches. The update process of the firmware itself will be covered by ISTG-FW[UPDT]. However, it must also be verified that software packages, which are running on the device and listening on interfaces, are up-to-date as well.

Usage of Outdated Software (ISTG-INT-CONF-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software is potentially vulnerable to attacks. For example, coding errors could lead to undefined program behavior, which then can be exploited by an attacker to gain access to data, processed by the application, or to perform actions in the context of the runtime environment. Furthermore, vulnerabilities in the used frameworks, libraries and other technologies might also affect the security level of a given piece of software.

Usually, developers release an update once a vulnerability was detected in their software. These updates should be installed as soon as possible in order to reduce the probability of successful attacks. Otherwise, attackers could use known vulnerabilities to perform attacks against the device.

Test Objectives

- The version identifiers of installed software packages as well as libraries and frameworks in use must be determined.
- Based on the detected version identifiers, it must be determined if the software version in use is up-to-date, e.g., by consulting the website of the software developer or public repositories.
- By using vulnerability databases, such as the National Vulnerability Database of the NIST, it has to be checked whether any vulnerabilities are known for the detected software versions.

Remediation

No outdated software packages should be running on the device. A proper patch management process, which ensures that applicable updates are installed once being available, should be implemented.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-001.

Presence of Unnecessary Software and Functionalities (ISTG-INT-CONF-002)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software, which is available on the device, broadens the attack surface since it might be used to perform attacks against the device. Even if the installed software is up-to-date, it might still be affected by unpublished vulnerabilities. It is also possible that a software program facilitates an attack without being vulnerable, e.g., by providing access to specific files or processes.

Test Objectives

- A list of functionalities, available via the interface, should be assembled.

- Based on the device documentation, its behavior and the intended use cases, it must be determined whether any of the available functionalities are not mandatory for the device operation.

Remediation

The attack surface should be minimized as much as possible by removing or disabling every software that is not required for the device operation.

Especially in case of general-purpose operating systems, such as Windows and Linux systems, it must be ensured that any unnecessary operating system features are disabled.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta

- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-002.

Secrets (ISTG-INT-SCRT)

IoT devices are often operated outside of the control space of their manufacturer. Still, they need to establish connections to other network nodes within the IoT ecosystem, e.g., to request and receive firmware updates or to send data to a cloud API. Hence, it might be required that the device has to provide some kind of authentication credential or secret. These secrets need to be stored on the device in a secure manner to prevent them from being stolen and used to impersonate the device.

Access to Confidential Data (ISTG-INT-SCRT-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Malfunctions, unintended behavior or improper implementation of an internal interface might enable an attacker to get access to secrets.

Test Objectives

- It has to be determined whether secrets can be accessed via the internal interface.

Remediation

Access to secrets should only be granted to individuals and processes that need to have access to them. No unauthorized or not properly authorized individual should be able to access secrets.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta

- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-SCRT-001.

Cryptography (ISTG-INT-CRYPT)

Many IoT devices need to implement cryptographic algorithms, e.g., to securely store sensitive data, for authentication purposes or to receive and verify encrypted data from other network nodes. Failing to implement secure, state of the art cryptography might lead to the exposure of sensitive data, device malfunctions or loss of control over the device.

Usage of Weak Cryptographic Algorithms (ISTG-INT-CRYPT-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Cryptography can be implemented in various ways. However, due to evolving technologies, new algorithms and more computing power becoming available, many old cryptographic algorithms are nowadays considered weak or insecure. Thus, either new and stronger cryptographic algorithms have to be used or existing algorithms must be adapted, e.g., by increasing the key length or using alternative modes of operation.

The usage of weak cryptographic algorithms might allow an attacker to recover the plaintext from a given ciphertext in a timely manner.

Test Objectives

- The data, processed by the interface, must be checked for the presence of encrypted data segments. In case that encrypted data segments are found, it must be checked whether the cryptographic algorithms in use can be identified.

- Furthermore, based on ISTG-INT-INFO-001, it must be checked whether headers, system messages etc. disclose the usage of certain cryptographic algorithms.

- In case that cryptographic algorithms can be identified, it must be determined whether the algorithms in use and their configuration are providing a sufficient level of security at the time of testing, e.g., by consulting cryptography guidelines like the technical guideline [TR-02102-1] (https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf __blob=publicationFile&v=10) by the BSI.

Remediation

Only strong, state of the art cryptographic algorithms should be used. Furthermore, these algorithms must be used in a secure manner by setting proper parameters, such as an appropriate key length or mode of operation.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CRYPT-001.

Business Logic (ISTG-INT-LOGIC)

Even if all other aspects of the internal interface are securely implemented and configured, issues in the underlying logic itself might render the device vulnerable to attacks. Thus, it must be verified if the internal interface and its functionalities are working as intended and if exceptions are detected and properly handled.

Circumvention of the Intended Business Logic (ISTG-INT-LOGIC-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Flaws in the implementation of the business logic might result in unintended behavior or malfunctions of the device. For example, if an attacker intentionally misses to provide relevant input data or tries to skip or change important steps in the processing workflow the device might end up in an unknown, potentially insecure state.

Test Objectives

- Based on the specific business logic implementation, it has to be determined whether deviations from the defined workflows are properly detected and handled.

Remediation

The device should not end up in an unknown state. Anomalies in the workflow must be detected and exceptions have to be handled properly.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-LOGIC-001.

Input Validation (ISTG-INT-INPV)

In order to ensure that only valid and well-formed data enters the processing flows of a device, the input from a all untrustworthy sources, e.g., users or external systems, has to be verified and validated.

Insufficient Input Validation (ISTG-INT-INPV-001)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place an attacker might be able to submit arbitrary and malformed data. Thus, the process, which handles the user input, or another downstream component might stop working properly due to not being able to process the data. This could result in malfunctions that might enable an attacker to manipulate the device behavior or render it unavailable.

Test Objectives

- It must be determined whether input to the internal interface is validated.
- In case that an input validation mechanism is implemented, it hast to be checked if there is a way to submit data, which does not comply with the intended data structure and value ranges.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-INPV-001.

Code or Command Injection (ISTG-INT-INPV-002)

Required Access Levels

Physical	*PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place an attacker might be able to submit code or commands, which then might be executed by the system. It strictly depends on the specific implementation of the device and the internal interface which code and commands are potentially executable. For example, a possible injection attack is OS command injection.

Test Objectives

- Based on ISTG-INT-INPV-001, it must be checked whether it is possible to submit code or commands, which are then executed by the system.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-INPV-002.

3.6. Physical Interfaces (ISTG-PHY)

Table of Contents

Overview

This section includes test cases and categories for the component physical interface. Depending on whether the interface is connected to a network or not, it might be accessible with *PA-2*, *PA-3* or *PA-4*. Establishing a direct connection to a physical interface might require specific hardware equipment (e.g., a connector or adapter cable).

In regards of test case categories that are relevant for a physical interface, the following were identified:

- **Authorization:** Focuses on vulnerabilities that allow to get unauthorized access to the physical interface or to elevate privileges in order to access restricted functionalities.

- **Information Gathering:** Focuses on information that is handled by the physical interface and that might be disclosed to potential attackers if not being properly protected or removed.
- **Configuration and Patch Management:** Focuses on vulnerabilities and issues in the configuration of a physical interface and its software components.
- **Secrets:** Focuses on secrets that are handled by the physical interface in an insecure manner.
- **Cryptography:** Focuses on vulnerabilities in the cryptographic implementation.
- **Business Logic:** Focuses on vulnerabilities in the implementation of the physical interface.
- **Input Validation:** Focuses on vulnerabilities regarding the validation and processing of input from untrustworthy sources.

Authorization (ISTG-PHY-AUTHZ)

Depending on the access model for a given device, only certain individuals might be allowed to access a physical interface. Thus, proper authentication and authorization procedures need to be in place, which ensure that only authorized users can get access.

Unauthorized Access to the Interface (ISTG-PHY-AUTHZ-001)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1*

Summary

Depending on the specific implementation of a given device, access to a physical interface might be restricted to individuals with a certain authorization access level, e.g., *AA-2*, *AA-3* or *AA-4*. If the device fails to correctly verify access permissions, any attacker (*AA-1*) might be able to get access.

Test Objectives

- It must be checked if authorization checks for access to the physical interface are implemented.

- In case that authorization checks are in place, it must be determined whether there is a way to bypass them.

Remediation

Proper authorization checks need to be implemented, which ensure that access to the physical interface is only possible for authorized individuals.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-AUTHZ-001.

Privilege Escalation (ISTG-PHY-AUTHZ-002)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-2 - AA-3* (depending on the access model for the given device)

Summary

Depending on the specific implementation of a given device, access to some functionalities via a physical interface might be restricted to individuals with a certain authorization access level, e.g., *AA-3* or *AA-4*. If the interface fails to correctly verify access permissions, an attacker with a lower authorization access level than intended might be able to get access to the restricted functionalities.

Test Objectives

- Based on ISTG-PHY-AUTHZ-001, it must be determined whether there is a way to elevate the given access privileges and thus to access restricted functionalities.

Remediation

Proper authorization checks need to be implemented, which ensure that access to restricted functionalities is only possible for individuals with the required authorization access levels.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-AUTHZ-002.

Information Gathering (ISTG-PHY-INFO)

Physical interfaces might disclose various information, which could reveal details regarding the inner workings of the device or the surrounding IoT ecosystem to potential attackers. This could enable and facilitate further, more advanced attacks.

Disclosure of Implementation Details (ISTG-PHY-INFO-001)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If details about the implementation, e.g., algorithms in use or the authentication procedure, are available to potential attackers, flaws and entry points for successful attacks are easier to detect. While the disclosure of such details alone is not considered to be a vulnerability, it facilitates the identification of potential attack vectors, thus allowing an attacker to exploit insecure implementations faster.

For example, relevant information might be included in service banners, response headers or error messages.

Test Objectives

- Accessible details regarding the implementation must be assessed in order to prepare further tests. For example, this includes:
  - Cryptographic algorithms in use
  - Authentication and authorization mechanisms
  - Local paths and environment details

Remediation

As mentioned above, the disclosure of such information is not considered a vulnerability. However, in order to impede exploitation attempts, only information, necessary for the device operation, should be displayed.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-002.

Disclosure of Ecosystem Details (ISTG-PHY-INFO-002)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

A physical interface might disclose information about the surrounding IoT ecosystem, e.g., sensitive URLs, IP addresses, software in use etc. An attacker might be able to use this information to prepare and execute attacks against the ecosystem.

For example, relevant information might be included in service banners, response headers or error messages.

Test Objectives

- It must be determined if the physical interface discloses relevant information about the surrounding ecosystem.

Remediation

The disclosure of information should be reduced to the minimum, which is required for operating the device. The disclosed information it has to be assessed and all unnecessarily included data should be removed.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-003.

Disclosure of User Data (ISTG-PHY-INFO-003)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

During runtime, a device is accumulating and processing data of different kinds, such as personal data of its users. If this data is disclosed, an attacker might be able to get access to it.

Test Objectives

- It has to be checked whether user data can be accessed by unauthorized individuals.

Remediation

Access to user data should only be granted to individuals and processes that need to have access to it. No unauthorized or not properly authorized individual should be able to access user data.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW[INST]-INFO-001.

Configuration and Patch Management (ISTG-PHY-CONF)

Since IoT devices can have a long lifespan, it is important to make sure that the software, running on the device, is regularly updated in order to apply the latest security patches. The update process of the firmware itself will be covered by ISTG-FW[UPDT]. However, it must also be verified that software packages, which are running on the device and listening on interfaces, are up-to-date as well.

Usage of Outdated Software (ISTG-PHY-CONF-001)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software is potentially vulnerable to attacks. For example, coding errors could lead to undefined program behavior, which then can be exploited by an attacker to gain access to data, processed by the application, or to perform actions in the context of the runtime environment. Furthermore, vulnerabilities in the used frameworks, libraries and other technologies might also affect the security level of a given piece of software.

Usually, developers release an update once a vulnerability was detected in their software. These updates should be installed as soon as possible in order to reduce the probability of successful attacks. Otherwise, attackers could use known vulnerabilities to perform attacks against the device.

Test Objectives

- The version identifiers of installed software packages as well as libraries and frameworks in use must be determined.
- Based on the detected version identifiers, it must be determined if the software version in use is up-to-date, e.g., by consulting the website of the software developer or public repositories.
- By using vulnerability databases, such as the National Vulnerability Database of the NIST, it has to be checked whether any vulnerabilities are known for the detected software versions.

Remediation

No outdated software packages should be running on the device. A proper patch management process, which ensures that applicable updates are installed once being available, should be implemented.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-001.

Presence of Unnecessary Software and Functionalities (ISTG-PHY-CONF-002)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software, which is available on the device, broadens the attack surface since it might be used to perform attacks against the device. Even if the installed software is up-to-date, it might still be affected by unpublished vulnerabilities. It is also possible that a software program facilitates an attack without being vulnerable, e.g., by providing access to specific files or processes.

Test Objectives

- A list of functionalities, available via the interface, should be assembled.
- Based on the device documentation, its behavior and the intended use cases, it must be determined whether any of the available functionalities are not mandatory for the device operation.

Remediation

The attack surface should be minimized as much as possible by removing or disabling every software that is not required for the device operation.

Especially in case of general-purpose operating systems, such as Windows and Linux systems, it must be ensured that any unnecessary operating system features are disabled.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-002.

Secrets (ISTG-PHY-SCRT)

IoT devices are often operated outside of the control space of their manufacturer. Still, they need to establish connections to other network nodes within the IoT ecosystem, e.g., to request and receive firmware updates or to send data to a cloud API. Hence, it might be required that the device has to provide some kind of authentication credential or secret. These secrets need to be stored on the device in a secure manner to prevent them from being stolen and used to impersonate the device.

Access to Confidential Data (ISTG-PHY-SCRT-001)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Malfunctions, unintended behavior or improper implementation of a physical interface might enable an attacker to get access to secrets.

Test Objectives

- It has to be determined whether secrets can be accessed via the physical interface.

Remediation

Access to secrets should only be granted to individuals and processes that need to have access to them. No unauthorized or not properly authorized individual should be able to access secrets.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-SCRT-001.

Cryptography (ISTG-PHY-CRYPT)

Many IoT devices need to implement cryptographic algorithms, e.g., to securely store sensitive data, for authentication purposes or to receive and verify encrypted data from other network nodes. Failing to implement secure, state of the art cryptography might lead to the exposure of sensitive data, device malfunctions or loss of control over the device.

Usage of Weak Cryptographic Algorithms (ISTG-PHY-CRYPT-001)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Cryptography can be implemented in various ways. However, due to evolving technologies, new algorithms and more computing power becoming available, many old cryptographic algorithms are nowadays considered weak or insecure. Thus, either new and stronger cryptographic algorithms have to be used or existing algorithms must be adapted, e.g., by increasing the key length or using alternative modes of operation.

The usage of weak cryptographic algorithms might allow an attacker to recover the plaintext from a given ciphertext in a timely manner.

Test Objectives

- The data, processed by the interface, must be checked for the presence of encrypted data segments. In case that encrypted data segments are found, it must be checked whether the cryptographic algorithms in use can be identified.

- Furthermore, based on ISTG-PHY-INFO-001, it must be checked whether headers, system messages etc. disclose the usage of certain cryptographic algorithms.

- In case that cryptographic algorithms can be identified, it must be determined whether the algorithms in use and their configuration are providing a sufficient level of security at the time of testing, e.g., by consulting cryptography guidelines like the technical guideline [TR-02102-1] (https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines /TG02102/BSI-TR-02102-1.pdf __blob=publicationFile&v=10) by the BSI.

Remediation

Only strong, state of the art cryptographic algorithms should be used. Furthermore, these algorithms must be used in a secure manner by setting proper parameters, such as an appropriate key length or mode of operation.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CRYPT-001.

Business Logic (ISTG-PHY-LOGIC)

Even if all other aspects of the physical interface are securely implemented and configured, issues in the underlying logic itself might render the device vulnerable to attacks. Thus, it must be verified if the physical interface and its functionalities are working as intended and if exceptions are detected and properly handled.

Circumvention of the Intended Business Logic (ISTG-PHY-LOGIC-001)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Flaws in the implementation of the business logic might result in unintended behavior or malfunctions of the device. For example, if an attacker intentionally misses to provide relevant input data or tries to skip or change important steps in the processing workflow the device might end up in an unknown, potentially insecure state.

Test Objectives

- Based on the specific business logic implementation, it has to be determined whether deviations from the defined workflows are properly detected and handled.

Remediation

The device should not end up in an unknown state. Anomalies in the workflow must be detected and exceptions have to be handled properly.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-LOGIC-001.

Input Validation (ISTG-PHY-INPV)

In order to ensure that only valid and well-formed data enters the processing flows of a device, the input from a all untrustworthy sources, e.g., users or external systems, has to be verified and validated.

Insufficient Input Validation (ISTG-PHY-INPV-001)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place an attacker might be able to submit arbitrary and malformed data. Thus, the process, which handles the user input, or another downstream component might stop working properly due to not being able to process the data. This could result in malfunctions that might enable an attacker to manipulate the device behavior or render it unavailable.

Test Objectives

- It must be determined whether input to the physical interface is validated.

- In case that an input validation mechanism is implemented, it hast to be checked if there is a way to submit data, which does not comply with the intended data structure and value ranges.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-INPV-001.

Code or Command Injection (ISTG-PHY-INPV-002)

Required Access Levels

Physical	*PA-2 - PA-4* (depending on whether the interface is connected to a network or not)
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place an attacker might be able to submit code or commands, which then might be executed by the system. It strictly depends on the specific implementation of the device and the physical interface which code and commands are potentially executable. For example, a possible injection attack is OS command injection.

Test Objectives

- Based on ISTG-PHY-INPV-001, it must be checked whether it is possible to submit code or commands, which are then executed by the system.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-INPV-002.

3.7. Wireless Interfaces (ISTG-WRLS)

Table of Contents

Overview

This section includes test cases and categories for the component wireless interface. A wireless interface is accessible with *PA-2*, *PA-3* or *PA-4*. Establishing a connection to a wireless interface might require specific hardware equipment (e.g., a dongle or software-defined radio).

In regards of test case categories that are relevant for a wireless interface, the following were identified:

- **Authorization:** Focuses on vulnerabilities that allow to get unauthorized access to the wireless interface or to elevate privileges in order to access restricted functionalities.

- **Information Gathering:** Focuses on information that is handled by the wireless interface and that might be disclosed to potential attackers if not being properly protected or removed.
- **Configuration and Patch Management:** Focuses on vulnerabilities and issues in the configuration of a wireless interface and its software components.
- **Secrets:** Focuses on secrets that are handled by the wireless interface in an insecure manner.
- **Cryptography:** Focuses on vulnerabilities in the cryptographic implementation.
- **Business Logic:** Focuses on vulnerabilities in the implementation of the wireless interface.
- **Input Validation:** Focuses on vulnerabilities regarding the validation and processing of input from untrustworthy sources.

Authorization (ISTG-WRLS-AUTHZ)

Depending on the access model for a given device, only certain individuals might be allowed to access a wireless interface. Thus, proper authentication and authorization procedures need to be in place, which ensure that only authorized users can get access.

Unauthorized Access to the Interface (ISTG-WRLS-AUTHZ-001)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1*

Summary

Depending on the specific implementation of a given device, access to a wireless interface might be restricted to individuals with a certain authorization access level, e.g., *AA-2*, *AA-3* or *AA-4*. If the device fails to correctly verify access permissions, any attacker (*AA-1*) might be able to get access.

Test Objectives

- It must be checked if authorization checks for access to the wireless interface are implemented.
- In case that authorization checks are in place, it must be determined whether there is a way to bypass them.

Remediation

Proper authorization checks need to be implemented, which ensure that access to the wireless interface is only possible for authorized individuals.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-AUTHZ-001.

Privilege Escalation (ISTG-WRLS-AUTHZ-002)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-2 - AA-3* (depending on the access model for the given device)

Summary

Depending on the specific implementation of a given device, access to some functionalities via a wireless interface might be restricted to individuals with a certain authorization access level, e.g., *AA-3* or *AA-4*. If the interface fails to correctly verify access permissions, an attacker with a lower authorization access level than intended might be able to get access to the restricted functionalities.

Test Objectives

- Based on ISTG-WRLS-AUTHZ-001, it must be determined whether there is a way to elevate the given access privileges and thus to access restricted functionalities.

Remediation

Proper authorization checks need to be implemented, which ensure that access to restricted functionalities is only possible for individuals with the required authorization access levels.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-AUTHZ-002.

Information Gathering (ISTG-WRLS-INFO)

Wireless interface might disclose various information, which could reveal details regarding the inner workings of the device or the surrounding IoT ecosystem to potential attackers. This could enable and facilitate further, more advanced attacks.

Disclosure of Implementation Details (ISTG-WRLS-INFO-001)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If details about the implementation, e.g., algorithms in use or the authentication procedure, are available to potential attackers, flaws and entry points for successful attacks are easier to detect. While the disclosure of such details alone is not considered to be a vulnerability, it facilitates the identification of potential attack vectors, thus allowing an attacker to exploit insecure implementations faster.

For example, relevant information might be included in service banners, broadcasts or error messages.

Test Objectives

- Accessible details regarding the implementation must be assessed in order to prepare further tests. For example, this includes:
    - Cryptographic algorithms in use
    - Authentication and authorization mechanisms
    - Local paths and environment details

Remediation

As mentioned above, the disclosure of such information is not considered a vulnerability. However, in order to impede exploitation attempts, only information, necessary for the device operation, should be displayed.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-002.

Disclosure of Ecosystem Details (ISTG-WRLS-INFO-002)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

A wireless interface might disclose information about the surrounding IoT ecosystem, e.g., sensitive URLs, IP addresses, software in use etc. An attacker might be able to use this information to prepare and execute attacks against the ecosystem.

For example, relevant information might be included in service banners, broadcasts or error messages.

Test Objectives

- It must be determined if the wireless interface discloses relevant information about the surrounding ecosystem.

Remediation

The disclosure of information should be reduced to the minimum, which is required for operating the device. The disclosed information it has to be assessed and all unnecessarily included data should be removed.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-003.

Disclosure of User Data (ISTG-WRLS-INFO-003)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

During runtime, a device is accumulating and processing data of different kinds, such as personal data of its users. If this data is disclosed, an attacker might be able to get access to it.

Test Objectives

- It has to be checked whether user data can be accessed by unauthorized individuals.

Remediation

Access to user data should only be granted to individuals and processes that need to have access to it. No unauthorized or not properly authorized individual should be able to access user data.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW[INST]-INFO-001.

Configuration and Patch Management (ISTG-WRLS-CONF)

Since IoT devices can have a long lifespan, it is important to make sure that the software, running on the device, is regularly updated in order to apply the latest security patches. The update process of the firmware itself will be covered by ISTG-FW[UPDT]. However, it must also be verified that software packages, which are running on the device and listening on interfaces, are up-to-date as well.

Usage of Outdated Software (ISTG-WRLS-CONF-001)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software is potentially vulnerable to attacks. For example, coding errors could lead to undefined program behavior, which then can be exploited by an attacker to gain access to data, processed by the application, or to perform actions in the context of the runtime environment. Furthermore, vulnerabilities in the used frameworks, libraries and other technologies might also affect the security level of a given piece of software.

Usually, developers release an update once a vulnerability was detected in their software. These updates should be installed as soon as possible in order to reduce the probability of successful attacks. Otherwise, attackers could use known vulnerabilities to perform attacks against the device.

Test Objectives

- The version identifiers of installed software packages as well as libraries and frameworks in use must be determined.

- Based on the detected version identifiers, it must be determined if the software version in use is up-to-date, e.g., by consulting the website of the software developer or public repositories.

- By using vulnerability databases, such as the National Vulnerability Database of the NIST, it has to be checked whether any vulnerabilities are known for the detected software versions.

Remediation

No outdated software packages should be running on the device. A proper patch management process, which ensures that applicable updates are installed once being available, should be implemented.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-001.

Presence of Unnecessary Software and Functionalities (ISTG-WRLS-CONF-002)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Every piece of software, which is available on the device, broadens the attack surface since it might be used to perform attacks against the device. Even if the installed software is up-to-date, it might still be affected by unpublished vulnerabilities. It is also possible that a software program facilitates an attack without being vulnerable, e.g., by providing access to specific files or processes.

Test Objectives

- A list of functionalities, available via the interface, should be assembled.

- Based on the device documentation, its behavior and the intended use cases, it must be determined whether any of the available functionalities are not mandatory for the device operation.

Remediation

The attack surface should be minimized as much as possible by removing or disabling every software that is not required for the device operation.

Especially in case of general-purpose operating systems, such as Windows and Linux systems, it must be ensured that any unnecessary operating system features are disabled.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-002.

Secrets (ISTG-WRLS-SCRT)

IoT devices are often operated outside of the control space of their manufacturer. Still, they need to establish connections to other network nodes within the IoT ecosystem, e.g., to request and receive firmware updates or to send data to a cloud API. Hence, it might be required that the device has to provide some kind of authentication credential or secret. These secrets need to be stored on the device in a secure manner to prevent them from being stolen and used to impersonate the device.

Access to Confidential Data (ISTG-WRLS-SCRT-001)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Malfunctions, unintended behavior or improper implementation of a wireless interface might enable an attacker to get access to secrets.

Test Objectives

- It has to be determined whether secrets can be accessed via the wireless interface.

Remediation

Access to secrets should only be granted to individuals and processes that need to have access to them. No unauthorized or not properly authorized individual should be able to access secrets.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-SCRT-001.

Cryptography (ISTG-WRLS-CRYPT)

Many IoT devices need to implement cryptographic algorithms, e.g., to securely store sensitive data, for authentication purposes or to receive and verify encrypted data from other network nodes. Failing to implement secure, state of the art cryptography might lead to the exposure of sensitive data, device malfunctions or loss of control over the device.

Usage of Weak Cryptographic Algorithms (ISTG-WRLS-CRYPT-001)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Cryptography can be implemented in various ways. However, due to evolving technologies, new algorithms and more computing power becoming available, many old cryptographic algorithms are nowadays considered weak or insecure. Thus, either new and stronger cryptographic algorithms have to be used or existing algorithms must be adapted, e.g., by increasing the key length or using alternative modes of operation.

The usage of weak cryptographic algorithms might allow an attacker to recover the plaintext from a given ciphertext in a timely manner.

Test Objectives

- The data, processed by the interface, must be checked for the presence of encrypted data segments. In case that encrypted data segments are found, it must be checked whether the cryptographic algorithms in use can be identified.

- Furthermore, based on ISTG-WRLS-INFO-001, it must be checked whether headers, system messages etc. disclose the usage of certain cryptographic algorithms.

- In case that cryptographic algorithms can be identified, it must be determined whether the algorithms in use and their configuration are providing a sufficient level of security at the time of testing, e.g., by consulting cryptography guidelines like the technical guideline TR-02102-1 by the BSI.

Remediation

Only strong, state of the art cryptographic algorithms should be used. Furthermore, these algorithms must be used in a secure manner by setting proper parameters, such as an appropriate key length or mode of operation.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CRYPT-001.

Business Logic (ISTG-WRLS-LOGIC)

Even if all other aspects of the wireless interface are securely implemented and configured, issues in the underlying logic itself might render the device vulnerable to attacks. Thus, it must be verified if the wireless interface and its functionalities are working as intended and if exceptions are detected and properly handled.

Circumvention of the Intended Business Logic (ISTG-WRLS-LOGIC-001)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

Flaws in the implementation of the business logic might result in unintended behavior or malfunctions of the device. For example, if an attacker intentionally misses to provide relevant input data or tries to skip or change important steps in the processing workflow the device might end up in an unknown, potentially insecure state.

Test Objectives

- Based on the specific business logic implementation, it has to be determined whether deviations from the defined workflows are properly detected and handled.

Remediation

The device should not end up in an unknown state. Anomalies in the workflow must be detected and exceptions have to be handled properly.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-LOGIC-001.

Input Validation (ISTG-WRLS-INPV)

In order to ensure that only valid and well-formed data enters the processing flows of a device, the input from a all untrustworthy sources, e.g., users or external systems, has to be verified and validated.

Insufficient Input Validation (ISTG-WRLS-INPV-001)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place an attacker might be able to submit arbitrary and malformed data. Thus, the process, which handles the user input, or another downstream component might stop working properly due to not being able to process the data. This could result in malfunctions that might enable an attacker to manipulate the device behavior or render it unavailable.

Test Objectives

- It must be determined whether input to the wireless interface is validated.
- In case that an input validation mechanism is implemented, it hast to be checked if there is a way to submit data, which does not comply with the intended data structure and value ranges.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-INPV-001.

Code or Command Injection (ISTG-WRLS-INPV-002)

Required Access Levels

Physical	*PA-2 - PA-4*
Authorization	*AA-1 - AA-4* (depending on the access model for the given device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place an attacker might be able to submit code or commands, which then might be executed by the system. It strictly depends on the specific implementation of the device and the ireless interface which code and commands are potentially executable. For example, a possible injection attack is OS command injection.

Test Objectives

- Based on ISTG-WRLS-INPV-001, it must be checked whether it is possible to submit code or commands, which are then executed by the system.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- "IoT Pentesting Guide" by Aditya Gupta
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-INPV-002.

3.8. User Interfaces (ISTG-UI)

Table of Contents

Overview

This section includes test cases and categories for the component user interface. Based on its implementation and intended use, a user interface might be accessible with all physical access levels.

In regards to test case categories that are relevant for a user interface, the following were identified:

- **Authorization:** Focuses on vulnerabilities that allow to get unauthorized access to the user interface or to elevate privileges in order to access restricted functionalities.

- **Information Gathering:** Focuses on information that is handled by the user interface and that might be disclosed to potential attackers if not being properly protected or removed.
- **Configuration and Patch Management:** Focuses on vulnerabilities and issues in the configuration of a user interface and its software components.
- **Secrets:** Focuses on secrets that are handled by the user interface in an insecure manner.
- **Cryptography:** Focuses on vulnerabilities in the cryptographic implementation.
- **Business Logic:** Focuses on vulnerabilities in the implementation of the user interface.
- **Input Validation:** Focuses on vulnerabilities regarding the validation and processing of input from untrustworthy sources.

Authorization (ISTG-UI-AUTHZ)

Depending on the access model for a given device, only certain individuals might be allowed to access a user interface. Thus, proper authentication and authorization procedures need to be in place, which ensure that only authorized users can get access.

Unauthorized Access to the Interface (ISTG-UI-AUTHZ-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1*

Summary

Depending on the specific implementation of a given device, access to a user interface might be restricted to individuals with a certain authorization access level, e.g., *AA-2*, *AA-3* or *AA-4*. If the device fails to correctly verify access permissions, any attacker (*AA-1*) might be able to get access.

Test Objectives

- It must be checked if authorization checks for access to the user interface are implemented.

- In case that authorization checks are in place, it must be determined whether there is a way to bypass them.

Remediation

Proper authorization checks need to be implemented, which ensure that access to the user interface is only possible for authorized individuals.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- OWASP "Mobile Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-AUTHZ-001.

Privilege Escalation (ISTG-UI-AUTHZ-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-2 - AA-3* (depending-on-the-access-model-for-the-given-device)

Summary

Depending on the specific implementation of a given device, access to some functionalities via a user interface might be restricted to individuals with a certain authorization access level, e.g., *AA-3* or *AA-4*. If the interface fails to correctly verify access permissions, an attacker with a lower authorization access level than intended might be able to get access to the restricted functionalities.

Test Objectives

- Based on ISTG-UI-AUTHZ-001, it must be determined whether there is a way to elevate the given access privileges and thus to access restricted functionalities.

Remediation

Proper authorization checks need to be implemented, which ensure that access to restricted functionalities is only possible for individuals with the required authorization access levels.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- OWASP "Mobile Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-AUTHZ-002.

Information Gathering (ISTG-UI-INFO)

User interface might disclose various information, which could reveal details regarding the inner workings of the device or the surrounding ISTG-ecosystem to potential attackers. This could enable and facilitate further, more advanced attacks.

Disclosure of Implementation Details (ISTG-UI-INFO-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

If details about the implementation, e.g., algorithms in use or the authentication procedure, are available to potential attackers, flaws and entry points for successful attacks are easier to detect. While the disclosure of such details alone is not considered to be a vulnerability, it facilitates the identification of potential attack vectors, thus allowing an attacker to exploit insecure implementations faster.

For example, relevant information might be included in service banners, response headers or error messages.

Test Objectives

- Accessible details regarding the implementation must be assessed in order to prepare further tests. For example, this includes:
    - Cryptographic algorithms in use
    - Authentication and authorization mechanisms
    - Local paths and environment details

Remediation

As mentioned above, the disclosure of such information is not considered a vulnerability. However, in order to impede exploitation attempts, only information, necessary for the device operation, should be displayed.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- OWASP "Mobile Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-002.

Disclosure of Ecosystem Details (ISTG-UI-INFO-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

A user interface might disclose information about the surrounding ISTG-ecosystem, e.g., sensitive URLs, IP addresses, software in use etc. An attacker might be able to use this information to prepare and execute attacks against the ecosystem.

For example, relevant information might be included in service banners, response headers or error messages.

Test Objectives

- It must be determined if the user interface discloses relevant information about the surrounding ecosystem.

Remediation

The disclosure of information should be reduced to the minimum, which is required for operating the device. The disclosed information it has to be assessed and all unnecessarily included data should be removed.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- OWASP "Mobile Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-INFO-003.

Disclosure of User Data (ISTG-UI-INFO-003)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

During runtime, a device is accumulating and processing data of different kinds, such as personal data of its users. If this data is disclosed, an attacker might be able to get access to it.

Test Objectives

- It has to be checked whether user data can be accessed by unauthorized individuals.

Remediation

Access to user data should only be granted to individuals and processes that need to have access to it. No unauthorized or not properly authorized individual should be able to access user data.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- OWASP "Mobile Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW[INST]-INFO-001.

Configuration and Patch Management (ISTG-UI-CONF)

Since ISTG-devices can have a long lifespan, it is important to make sure that the software, running on the device, is regularly updated in order to apply the latest security patches. The update process of the firmware itself will be covered by ISTG-FW[UPDT]. However, it must also be verified that software packages, which are running on the device and listening on interfaces, are up-to-date as well.

Usage of Outdated Software (ISTG-UI-CONF-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

Every piece of software is potentially vulnerable to attacks. For example, coding errors could lead to undefined program behavior, which then can be exploited by an attacker to gain access to data, processed by the application, or to perform actions in the context of the runtime environment. Furthermore, vulnerabilities in the used frameworks, libraries and other technologies might also affect the security level of a given piece of software.

Usually, developers release an update once a vulnerability was detected in their software. These updates should be installed as soon as possible in order to reduce the probability of successful attacks. Otherwise, attackers could use known vulnerabilities to perform attacks against the device.

Test Objectives

- The version identifiers of installed software packages as well as libraries and frameworks in use must be determined.
- Based on the detected version identifiers, it must be determined if the software version in use is up-to-date, e.g., by consulting the website of the software developer or public repositories.
- By using vulnerability databases, such as the National Vulnerability Database of the NIST, it has to be checked whether any vulnerabilities are known for the detected software versions.

Remediation

No outdated software packages should be running on the device. A proper patch management process, which ensures that applicable updates are installed once being available, should be implemented.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-001.

Presence of Unnecessary Software and Functionalities (ISTG-UI-CONF-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

Every piece of software, which is available on the device, broadens the attack surface since it might be used to perform attacks against the device. Even if the installed software is up-to-date, it might still be affected by unpublished vulnerabilities. It is also possible that a software program facilitates an attack without being vulnerable, e.g., by providing access to specific files or processes.

Test Objectives

- A list of functionalities, available via the interface, should be assembled.
- Based on the device documentation, its behavior and the intended use cases, it must be determined whether any of the available functionalities are not mandatory for the device operation.

Remediation

The attack surface should be minimized as much as possible by removing or disabling every software that is not required for the device operation.

Especially in case of general-purpose operating systems, such as Windows and Linux systems, it must be ensured that any unnecessary operating system features are disabled.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CONF-002.

Secrets (ISTG-UI-SCRT)

ISTG-devices are often operated outside of the control space of their manufacturer. Still, they need to establish connections to other network nodes within the ISTG-ecosystem, e.g., to request and receive firmware updates or to send data to a cloud API. Hence, it might be required that the device has to provide some kind of authentication credential or secret. These secrets need to be stored on the device in a secure manner to prevent them from being stolen and used to impersonate the device.

Access to Confidential Data (ISTG-UI-SCRT-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

Malfunctions, unintended behavior or improper implementation of a user interface might enable an attacker to get access to secrets.

Test Objectives

- It has to be determined whether secrets can be accessed via the user interface.

Remediation

Access to secrets should only be granted to individuals and processes that need to have access to them. No unauthorized or not properly authorized individual should be able to access secrets.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- OWASP "Mobile Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-SCRT-001.

Cryptography (ISTG-UI-CRYPT)

Many ISTG-devices need to implement cryptographic algorithms, e.g., to securely store sensitive data, for authentication purposes or to receive and verify encrypted data from other network nodes. Failing to implement secure, state of the art cryptography might lead to the exposure of sensitive data, device malfunctions or loss of control over the device.

Usage of Weak Cryptographic Algorithms (ISTG-UI-CRYPT-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

Cryptography can be implemented in various ways. However, due to evolving technologies, new algorithms and more computing power becoming available, many old cryptographic algorithms are nowadays considered weak or insecure. Thus, either new and stronger cryptographic algorithms have to be used or existing algorithms must be adapted, e.g., by increasing the key length or using alternative modes of operation.

The usage of weak cryptographic algorithms might allow an attacker to recover the plaintext from a given ciphertext in a timely manner.

Test Objectives

- The data, processed by the interface, must be checked for the presence of encrypted data segments. In case that encrypted data segments are found, it must be checked whether the cryptographic algorithms in use can be identified.
- Furthermore, based on ISTG-UI-INFO-001, it must be checked whether headers, system messages etc. disclose the usage of certain cryptographic algorithms.
- In case that cryptographic algorithms can be identified, it must be determined whether the algorithms in use and their configuration are providing a sufficient level of security at the time of testing, e.g., by consulting cryptography guidelines like the technical guideline TR-02102-1 by the BSI.

Remediation

Only strong, state of the art cryptographic algorithms should be used. Furthermore, these algorithms must be used in a secure manner by setting proper parameters, such as an appropriate key length or mode of operation.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-FW-CRYPT-001.

Business Logic (ISTG-UI-LOGIC)

Even if all other aspects of the user interface are securely implemented and configured, issues in the underlying logic itself might render the device vulnerable to attacks. Thus, it must be verified if the user interface and its functionalities are working as intended and if exceptions are detected and properly handled.

Circumvention of the Intended Business Logic (ISTG-UI-LOGIC-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

Flaws in the implementation of the business logic might result in unintended behavior or malfunctions of the device. For example, if an attacker intentionally misses to provide relevant input data or tries to skip or change important steps in the processing workflow the device might end up in an unknown, potentially insecure state.

Test Objectives

- Based on the specific business logic implementation, it has to be determined whether deviations from the defined workflows are properly detected and handled.

Remediation

The device should not end up in an unknown state. Anomalies in the workflow must be detected and exceptions have to be handled properly.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- OWASP "Mobile Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-LOGIC-001.

Input Validation (ISTG-UI-INPV)

In order to ensure that only valid and well-formed data enters the processing flows of a device, the input from a all untrustworthy sources, e.g., users or external systems, has to be verified and validated.

Insufficient Input Validation (ISTG-UI-INPV-001)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place an attacker might be able to submit arbitrary and malformed data. Thus, the process, which handles the user input, or another downstream component might stop working properly due to not being able to process the data. This could result in malfunctions that might enable an attacker to manipulate the device behavior or render it unavailable.

Test Objectives

- It must be determined whether input to the user interface is validated.
- In case that an input validation mechanism is implemented, it hast to be checked if there is a way to submit data, which does not comply with the intended data structure and value ranges.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- OWASP "Mobile Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta

- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-INPV-001.

Code or Command Injection (ISTG-UI-INPV-002)

Required Access Levels

Physical	*PA-1 - PA-4* (depending-on-how-the-user-interface-can-be-accessed,-e.g.,-if-they-were-designed-for-remote-access)
Authorization	*AA-1 - AA-4* (depending-on-the-access-model-for-the-given-device)

Summary

If no input validation is performed or only an insufficient input validation mechanism is in place an attacker might be able to submit code or commands, which then might be executed by the system. It strictly depends on the specific implementation of the device and the user interface which code and commands are potentially executable. For example, possible injection attacks are Cross Site Scripting, SQL injection and OS command injection.

Test Objectives

- Based on ISTG-UI-INPV-001, it must be checked whether it is possible to submit code or commands, which are then executed by the system.

Remediation

The device has to validate all input from untrustworthy sources. Malformed or otherwise invalid input must either be rejected or converted into a proper data structure, e.g., by encoding the input. However, it must be ensured that the input is not interpreted or executed when converting it.

References

For this test case, data from the following sources was consolidated:

- OWASP "Web Security Testing Guide"
- OWASP "Mobile Security Testing Guide"
- "IoT Penetration Testing Cookbook" by Aaron Guzman and Aditya Gupta
- "The IoT Hacker's Handbook" by Aditya Gupta
- "Practical IoT Hacking" by Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, and Beau Woods
- Key aspects of testing of the T-Systems Multimedia Solutions GmbH

This test case is based on: ISTG-DES-INPV-002.

Project Collaborators and Acknowledgements

We would like to take this opportunity to acknowledge the contributions of our collaborators and supporters who volunteered their time and expertise to this project. Thank you for your support and commitment to IoT security! This guide would not have been possible without you.

- Antje Winkler
- Clemens Keil
- Denny Vogt (Pyxon73)
- Manfred Heinz (zaphoxx aka CptSpiff)
- Martin Weißbach
- Patrick "HomeSen" Walker
- Sebastian Döring

Attribution-ShareAlike 4.0 International

===

Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

```
  Considerations for licensors: Our public licenses are
  intended for use by those authorized to give the public
  permission to use material in ways otherwise restricted by
  copyright and certain other rights. Our licenses are
  irrevocable. Licensors should read and understand the terms
  and conditions of the license they choose before applying it.
  Licensors should also secure all rights necessary before
  applying our licenses so that the public can reuse the
  material as expected. Licensors should clearly mark any
  material not subject to the license. This includes other CC-
  licensed material, or material used under an exception or
  limitation to copyright. More considerations for licensors:
 wiki.creativecommons.org/Considerations_for_licensors

  Considerations for the public: By using one of our public
  licenses, a licensor grants the public permission to use the
  licensed material under specified terms and conditions. If
  the licensor's permission is not necessary for any reason--for
  example, because of any applicable exception or limitation to
  copyright--then that use is not regulated by the license. Our
  licenses grant only permissions under copyright and certain
  other rights that a licensor has authority to grant. Use of
  the licensed material may still be restricted for other
  reasons, including because others have copyright or other
  rights in the material. A licensor may make special requests,
  such as asking that all changes be marked or described.
  Although not required by our licenses, you are encouraged to
  respect those requests where reasonable. More considerations
  for the public:
 wiki.creativecommons.org/Considerations_for_licensees
```

===

Creative Commons Attribution-ShareAlike 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-ShareAlike 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 -- Definitions.

a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. BY-SA Compatible License means a license listed at creativecommons.org/compatiblelicenses, approved by Creative Commons as essentially the equivalent of this Public License.

d. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

e. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

f. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

g. License Elements means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution and ShareAlike.

h. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

i. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

j. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

k. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

l. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

m. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 -- Scope.

a. License grant.

```
1. Subject to the terms and conditions of this Public License,
   the Licensor hereby grants You a worldwide, royalty-free,
   non-sublicensable, non-exclusive, irrevocable license to
   exercise the Licensed Rights in the Licensed Material to:

     a. reproduce and Share the Licensed Material, in whole or
        in part; and

     b. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where
   Exceptions and Limitations apply to Your use, this Public
   License does not apply, and You do not need to comply with
   its terms and conditions.

3. Term. The term of this Public License is specified in Section
   6(a).

4. Media and formats; technical modifications allowed. The
   Licensor authorizes You to exercise the Licensed Rights in
   all media and formats whether now known or hereafter created,
   and to make technical modifications necessary to do so. The
   Licensor waives and/or agrees not to assert any right or
   authority to forbid You from making technical modifications
   necessary to exercise the Licensed Rights, including
   technical modifications necessary to circumvent Effective
   Technological Measures. For purposes of this Public License,
   simply making modifications authorized by this Section 2(a)
   (4) never produces Adapted Material.

5. Downstream recipients.

     a. Offer from the Licensor -- Licensed Material. Every
        recipient of the Licensed Material automatically
        receives an offer from the Licensor to exercise the
        Licensed Rights under the terms and conditions of this
        Public License.

     b. Additional offer from the Licensor -- Adapted Material.
        Every recipient of Adapted Material from You
        automatically receives an offer from the Licensor to
        exercise the Licensed Rights in the Adapted Material
        under the conditions of the Adapter's License You apply.

     c. No downstream restrictions. You may not offer or impose
        any additional or different terms or conditions on, or
        apply any Effective Technological Measures to, the
        Licensed Material if doing so restricts exercise of the
        Licensed Rights by any recipient of the Licensed
        Material.

6. No endorsement. Nothing in this Public License constitutes or
   may be construed as permission to assert or imply that You
   are, or that Your use of the Licensed Material is, connected
   with, or sponsored, endorsed, or granted official status by,
   the Licensor or others designated to receive attribution as
   provided in Section 3(a)(1)(A)(i).
```

b. Other rights.

```
1. Moral rights, such as the right of integrity, are not
   licensed under this Public License, nor are publicity,
   privacy, and/or other similar personality rights; however, to
   the extent possible, the Licensor waives and/or agrees not to
   assert any such rights held by the Licensor to the limited
   extent necessary to allow You to exercise the Licensed
   Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this
   Public License.

3. To the extent possible, the Licensor waives any right to
   collect royalties from You for the exercise of the Licensed
   Rights, whether directly or through a collecting society
   under any voluntary or waivable statutory or compulsory
   licensing scheme. In all other cases the Licensor expressly
   reserves any right to collect such royalties.
```

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

```
1. If You Share the Licensed Material (including in modified
   form), You must:

     a. retain the following if it is supplied by the Licensor
        with the Licensed Material:

          i. identification of the creator(s) of the Licensed
             Material and any others designated to receive
             attribution, in any reasonable manner requested by
             the Licensor (including by pseudonym if
             designated);

         ii. a copyright notice;

        iii. a notice that refers to this Public License;

         iv. a notice that refers to the disclaimer of
             warranties;

          v. a URI or hyperlink to the Licensed Material to the
             extent reasonably practicable;

     b. indicate if You modified the Licensed Material and
        retain an indication of any previous modifications; and

     c. indicate the Licensed Material is licensed under this
        Public License, and include the text of, or the URI or
        hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any
   reasonable manner based on the medium, means, and context in
   which You Share the Licensed Material. For example, it may be
   reasonable to satisfy the conditions by providing a URI or
   hyperlink to a resource that includes the required
   information.

3. If requested by the Licensor, You must remove any of the
   information required by Section 3(a)(1)(A) to the extent
   reasonably practicable.
```

b. ShareAlike.

```
In addition to the conditions in Section 3(a), if You Share
Adapted Material You produce, the following conditions also apply.

  1. The Adapter's License You apply must be a Creative Commons
     license with the same License Elements, this version or
     later, or a BY-SA Compatible License.

  2. You must include the text of, or the URI or hyperlink to, the
     Adapter's License You apply. You may satisfy this condition
     in any reasonable manner based on the medium, means, and
     context in which You Share Adapted Material.

  3. You may not offer or impose any additional or different terms
     or conditions on, or apply any Effective Technological
     Measures to, Adapted Material that restrict exercise of the
     rights granted under the Adapter's License You apply.
```

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material, including for purposes of Section 3(b); and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 -- Disclaimer of Warranties and Limitation of Liability.

a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 -- Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

```
    1. automatically as of the date the violation is cured, provided
       it is cured within 30 days of Your discovery of the
       violation; or

    2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any
right the Licensor may have to seek remedies for Your violations
of this Public License.
```

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 -- Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

===

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the “Licensor.” The text of the Creative Commons public licenses is dedicated to the public domain under the CC0 Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.