Federal Office
for Information Security

Technical Guideline TR-03161: Requirements for Healthcare Applications

Part 1: Mobile Applications

Version 3.0

Change history

Version	*Date*	*Name*	*Description*
3.0	11.09.2024	Unit D 24	Translation of German version 3.0

Federal Office for Information Technology
Postfach 20 03 63
53133 Bonn
Tel: + 49 22899 9582-0
E-Mail: referat-d24@bsi.bund.de
Web: https://www.bsi.bund.de

Table of Contents

List of tables

1 Introduction

1.1 Scope of the Technical Guideline

Health applications (e-health) follow the overall goal of supporting the treatment and care of patients and utilizing the opportunities offered by modern information and communication technologies (IKT) (see [BMG-EH]). In the context of this Technical Guideline (TR), digital health applications and digital care applications should be highlighted in particular:

According to Section § 33a of the German Social Code Book V (SGB V), those with German statutory health insurance have, under certain conditions, the right to the supply of so-called digital health applications.

According to Section § 40a of the German Social Code Book XI (SGB XI), patients in social care insurance have, under certain conditions, the right to the supply of so-called digital care applications.

The TR is aimed at developers of healthcare applications for mobile devices. In addition, it can be considered as a directive for mobile applications that process or store sensitive data.

1.2 Goal of the Technical Guideline

Digitalization of all areas of life, whether at work, in home environments, in individual or public transportation, is progressing steadily. Already in 2018, the number of internet users exceeded the limit of 4 billion people. Two thirds of the world's population, currently counting 7.6 billion people, use a mobile phone. More than three billion people use social media and do so in nine out of ten cases via their smartphones (see [GDR18]). This development continues in the health care sector. Beginning with the trend of "self-tracking", but also with the increasing demand for the efficient use of collected medical data. Especially in the health care sector, it is comfortable that your own medical data can be accessible regardless of your current location and time. In these circumstances, backend systems store sensitive and personal data, from pulse frequency, sleep rhythm records to medication plans and medical prescriptions. Backend systems connect the user with multiple services and therefore act as communication hubs. A compromised mobile device can disclose the entire digital life of the user unintentionally, which may lead to high financial damage. Compliance with appropriate security standards, especially in the area of backend systems, can decrease the risk and may even prevent it at all. Already during the development phase, manufacturers should plan very responsibly how a backend system processes, stores and protects personal, in this case medical and other sensitive data.

For protecting sensitive data and processes, IT security follows three main protection objectives: confidentiality, integrity and availability.

Compliance with these three main objectives, is of particular importance for healthcare applications. Once the users' health data is unlawfully disclosed, the confidentiality for these data and the users' health status is forever lost. While the concerned user could receive compensation for this purpose, the publication cannot be reversed.

In addition, unintentional publication of health data, in social and professional spheres, can lead to incalculable consequences.

An attacker, gaining access to these kind of sensitive health data, can be able to manipulate the users' health data and thus harm its integrity. This data manipulation may have a significant impact on treatment decisions and ultimately on the users' health. In addition to the manipulation of medical data, manipulation of the entire application must be considered a risk to such applications, as this may misrepresent the indication for the user.

This Technical Guideline should therefore serve as a guide to assist developers of mobile applications in creating secure mobile applications for healthcare applications. If the applications relies on functionalities of a backend systems, the security of the backend system is also essential for a full safety assessment (see chapter 2.3).

1.3 Overview of the Technical Directive

1.3.1 Methodology

Applications within the scope of this TR are applications on (mobile) devices, like smartphones. This includes in particular digital health applications within the meaning of § 33a SGB V (see [SGBV33a]) and mobile digital care applications in accordance with § 40a SGB XI (see [SGBXI40a]). The usage can be implemented autonomously, therefore only depending on the application on the users' device itself or in combination with a secure backend system. The term 'backend system' within this guideline refers in particular to the use of cloud computing. Due to rapid technological progress and the diversity of architectures and configuration possibilities of mobile devices, this Technical Guideline does not claim completeness. Instead, it can be considered as a minimum requirement for the secure operation of an application.

The Technical Guideline formulates a security problem definition (SPD) which has potential threat scenarios. From this SPD, testing aspects for backend systems and their platforms or operational environments are derived to protect against threats.

The threat scenarios and test aspects formulated in this Technical Guideline are based on experience gained by the BSI in previous inspections of health care backend systems. Additionally, it follows international standards, such as the Smartphone Secure Development Guidelines [SSDG] and the Mobile AppSec Verification Standard [MASVS], with its accompanying Mobile Security Testing Guide [MSTG].

A basic requirement for applications within the scope of the Technical Guideline is guidance on best practice recommendations and other general requirements for secure, distributed applications. These include carrying out intensive functional tests, integration tests and, in particular, test the applications' behavior during expected and unexpected (user) inputs (positive/negative tests). The TR also imposes additional, specific requirements.

1.3.2 Terms

This Technical Directive uses the following terms:

Table 1: Concepts of the Technical Directive

Term	Description
MUST	The application must necessarily have a certain property.
MUST NOT	The application must under no circumstances have a particular property.
SHOULD	The application must have a specific property, except that non-implementation does not present a risk to safe operation or is currently not possible due to technical constraints.
MAY	The application may have a specific property, with a translation of that property to be indicated by the solution provider.

Term	Description
primary purpose	The primary purpose of the application within the Technical Guideline is a purpose of the intended use and any purposes directly following the legal framework. (For digital health applications under Section § 33a SGB V, the purposes under Section § 4(2) sentence 1 (1) to (3) [DIGAV] together with the obligations under medical device law form the primary purpose.)
lawful purpose	The legitimate purpose of an application within the Technical Guideline is a purpose permitted by law as a basis for processing personal data. (For digital health applications under Section § 33a SGB V, these purposes are defined in the statutory order in Section § 4(2) and (4) [DIGAV].

2 Overview of Security Requirements for Healthcare Applications

2.1 Application concepts on mobile terminals

The term “mobile application” describes a program executed on a mobile device. In principle, such applications can be divided into three categories. The first category is the native applications (chapter 2.1.1), which are directly tailored to the platform on which they are executed. This is compared with web applications (chapter 2.2). Their implementation is completely independent of the platform and they run within the mobile devices’ web browser. The third category includes hybrid approaches (chapter 2.1.2). They combine all possible combinations of native applications and web applications.

As web applications go far beyond the general use on mobile devices, this publication focuses on native applications and the native part of hybrid approaches. For additional guidance on the safe development and operation of web applications, the BSI recommends “TR-03161 Requirements for Healthcare Applications Part 2: Web applications” [TR03161-2].

2.1.1 Native applications

A native application is tailored to a platform and its operating system. It is based on the Software Development Kits (SDKs) provided by the platform (e.g. Android or iOS). These SDKs allow direct access to device components such as GPS, camera or microphone by the application. These applications benefit from the closeness to the operating system, therefore can achieve very good performance, high reliability and intuitive operability. Native applications can be installed via the platform-owned app stores and can often be used offline as well.

However, there are also disadvantages associated with the closeness to the operating system. Changes to the operating system, for example through updates, may require adjustments to the application as well. Not updating the application may lead to usage restrictions. Moreover, it is not possible to install native applications on other operating systems. If the same application is to be published on several operating systems, there must be separate code[1] bases, which is often a high expenditure.

2.1.2 Hybrid approaches

Hybrid applications combine both the advantages and disadvantages of native applications, as well as web applications and web services. The SDK provides a framework application that has all the advantages and disadvantages of native applications. It can access device components and be installed via a OS-specific app store, but not installed on other platforms without making adjustments to the source code. In addition, the framework applications include an embedded web browser that allows web applications to be integrated into the native framework applications. This also allows web applications to access the device components otherwise reserved for native applications only. In addition, the use of different user interfaces may negatively impact user experience. The platform dependency of the application now refers only to the framework application, which significantly reduces the expenditure on migrating to other platforms.

2.2 Web applications

Web applications are applications, mostly web pages, which can be run in combination with a backend system (chapter 2.3) without installation on a local system. Such websites are often programmed to look like

[1] There are also cross-platform implementation approaches that support the development of an application for different platforms at the same time. However, this only shifts dependency to this very extensive middleware, which must cover all target platforms.

a native application for classical desktop systems or mobile devices and to behave in a comparable way. Unlike native applications, they are not based on an SDK of the underlying platforms, but on classic web development programming tools. In most cases HTML5 and JavaScript are used. For this reason, they can only have very limited access to device components. Their main advantage is that they are independent of the operating system. As the applications run within a web browser, they can be used equally on any platform without having to adapt to the code base.

As this publication focuses on mobile applications, the following threat analyses and audit aspects relate to the protection of mobile applications. For further guidance on secure operation and development of web applications, we refer to "TR-03161 Requirements for Healthcare Applications Part 2: Web applications" [TR03161-2].

2.3 Backend systems

Most applications do not rely exclusively on the resources provided by the users' device to process and store data. They will instead outsource these tasks on a server system. Because these servers are not visible from a user's perspective, they are also called backend systems or backend services (as a delimitation to the application that the user sees, which is defined as front end). In addition to specialized processing and storage of data, these systems often perform user authentication and authorization tasks or other central activities. Therefore, not all functionalities of the applications have to be implemented on the users' device, which are often limited to the user interface. A general statement about how much functionality is implemented in the application itself and how much is outsourced to a server cannot be made. Depending on the individual applications, the amount of outsourced tasks vary. Nevertheless, when considering the security of the whole healthcare application, evaluating the security of the backend system is essential.

For the use of applications connected to a backend system, an active internet connection is most often mandatory. A transport connection secured through TLS is usually used for communication between the front and the backend system. The use of backend systems is not limited to the area of mobile applications, but reflects the state of the art for almost all applications. Three main scenarios for using backend systems are distinguished:

- The manufacturer of the application manages the infrastructure of the backend system itself (chapter 2.3.1).
- The manufacturer of the application has the infrastructure managed by an external service provider (chapter 2.3.2).
- The entire backend system of the application is hosted by a cloud service provider (chapter 2.3.3).

Depending on the type of operation and associated different attack vectors, the application manufacturer has different possibilities to ensure the security of the overall application and of the data stored and processed. As this publication focuses on mobile applications, the following threat analyses and audit aspects relate to the protection of the mobile application. For further guidance on secure operation and development of backend systems, we refer to "TR-03161 Requirements for Healthcare Applications Part 3: Backend systems" [TR03161-3].

2.3.1 Self-hosted systems

In the case of self-hosted systems, the manufacturer of the application acts itself as the operator of the backend systems. It thus has direct access to the systems and environment. The server running the backend is located within the manufacturer's operational environment and the physical, technical and organizational security measurements of the systems are provided by the manufacturer itself. The main advantage of this solution is, the manufacturer keeps sole sovereignty over all systems and is able to react quickly and directly to any process/event. As the manufacturer manages the systems itself and selects/develops all software components, it has the greatest knowledge of potential vulnerabilities of these systems. However, in this case, it also imposes sole responsibility on the manufacturer for providing

permanent staff for monitoring and responding to any incidents and ensuring the usability of the application. Depending on the manufacturer's business direction, the manufacturers' expertise may be within developing healthcare applications, whether providing IT security for the whole operations life cycle.

2.3.2 Externally hosted systems

In the case of externally hosted systems, servers used by the application are hosted by an external service provider, which is most likely specialized in hosting. The security benefits of the service providers' higher experience in operating backend systems and therefore having a positive impact on the applications' availability. Depending on the individual hosting agreement, the service provider performs additional tasks, as in providing security updates for the operation systems, performing backups or monitoring the system, to respond to suspicious activities immediately.

The manufacturer has to trust the service provider to a certain degree, due to the lack of sovereignty. For example, the manufacturer cannot monitor the integrity of the servers' hardware, even though, direct physical access can always evade software monitoring measures. In addition, the service provider has many costumers, all of whom need to be separated at a technical level, in order to avoid information leakages to competitors or the general public. Due to the shared responsibility between manufacturer and the service provider, friction losses may occur, costing valuable time, especially in critical situations.

2.3.3 Cloud computing

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computer resources (e.g. networks, servers, storage systems, applications and services) that can be provisioned rapidly and released with minimum management effort or service provider interaction. Resources may be extended flexibly depending on the customer's needs. As a result, the provider of the service has even less influence on the service's environment than with a simple external hosting. It is no longer possible, for example, to identify on which device exactly a particular operation is carried out. The manufacturer relies fully on the cloud service provider. Therefore, when using cloud computing for applications within the meaning of the TR, cloud provider fulfilling the requirements of the BSI's "C5 criteria catalogue" should [KCC-C5] only be used. The manufacturer must verify, on the basis of the submitted C5 attestation, whether the requirements of the TR are met by the cloud service used. As an alternative to the C5 testate, cloud providers with comparable attestations or certificates are also allowed (see [TR03161-3]).

2.4 Security problem definition

The Security Problem Definition describes assumptions, threats and organizational security policies necessary to deliver security for healthcare applications.

2.4.1 Assumptions

A.Device — The mobile device running the application is operated by the user itself and protected from vulnerabilities, this includes for example keeping the operating system updated. The safety of the mobile device has not been compromised due to changes made by the user. Devices that are no longer supplied with security updates by the manufacturer shall not be used.

A.Source — The application and its updates are obtained exclusively from secure sources that the developer has approved for publication (e.g. app stores, proprietary website, public bodies such as competent authorities and health insurances). The installed applications are regularly checked for updates and updated.

A.Backend The backend system is located in a protected environment. Organizational and technical measures ensure that attackers cannot gain physical access to the infrastructure of the backend system. The backend system meets the requirements of “TR-03161 Requirements for Healthcare Applications Part 3: Backend systems” [TR03161-3].

A.OperatingSystem Only the functions and components provided by the operating system that are required for the legitimate purpose of the application are used. The functions used by means of the authorizations granted by the operating system do not increase the attack surface and are assumed to be secure. This applies in particular to cryptographic functions and protocols. The operating system does not share sensitive application data with third parties without the user's prior consent.

2.4.2 Threats

T.SensitiveData Sensitive data in the Technical Guideline are as defined in Appendix A are to be understood. An unauthorized person gains access to such sensitive data in the application, such as sensitive notifications in the lock screen, unencrypted data stored in the file system or RAM. This also means that an attacker can access encrypted, sensitive data in plain text after analyzing the encryption mechanism.

T.Auth An attacker gains access to sensitive data of other users using a different user ID, role or group membership.

T.Eavesdropping An attacker is able to eavesdrop on the user (e.g. via an insufficiently encrypted/non-authenticated connection) using insufficiently authenticated inter-process communication (IPC) calls[2]. For example, an attacker can exploit the establishment of a transport connection to gain access to sensitive data due to a lack of verification of certificate properties.

T.DevFunctions An attacker uses hidden or remaining developers or debugging functions in the application to subvert security measures.

T.Expense The application causes unforeseen, additional costs for the user or operator.

T.Impersonation An attacker gains unauthorized access to sensitive data or chargeable functions of another user through missing or faulty access controls or by guessing access parameters.

T.InfoDisclosure An attacker analyzes the application and finds references for e.g. hard-coded test accounts or cryptographic secrets.

T.Integrity An attacker is able to manipulate or delete data within memory or via the transport route without being noticed.

T.MemoryStructures An attacker performs reverse engineering on the application and thereby identifies unprotected data structures in the memory, enabling access to keys and sensitive data.

T.VisibleAsset The attacker can read sensitive data displayed on the application by “shoulder surfing”[3].

[2] IPC (*Inter Process Communication*) is used on mobile devices as a communication channel between different applications.

[3] While shoulder surfing, the attacker looks over the shoulder of the user unnoticed to obtain information.

2.4.3 Organizational security policies

OSP.Authorization The manufacturer develops an authorization concept that controls both read and write access to sensitive data. The access authorizations must be selected in such a way that only the rights required to fulfil the primary or legitimate purpose are granted. The authorization concept must be implemented independently of authentication.

OSP.Biometry If biometrics is used for authentication, the suitability of the platform and the stored reference value must be verified before each application session.

OSP.CriticalUpdates The manufacturer permanently checks and monitors the application and the third-party software used [4] for exploitable vulnerabilities. The developer must provide an update at short notice if vulnerabilities become known. The backend system must inform the application about the update and prevent the use of the application in an outdated version after a transition period (grace period).

OSP.DataSovereignty The application ensures the user's data sovereignty. The application informs the user of existing risks due to the configuration of their mobile device and allows them to decide to cancel use. At the user's request, the application deletes data already recorded on all local storage media. In case the app is connected to a backend system, the app asks the backend system to delete the data.

OSP.Disclosure The developer offers a low-threshold process for reporting vulnerabilities. This means that it provides easy-to-find contact information for the security department and offers a way to report vulnerabilities anonymously.

OSP.LibsIn Incoming data from third-party software should be validated before being used in the application (e.g. XML schema validation, checking for invalid encoding, etc.). The aim is to protect the application from attacks by malicious input.

OSP.LibsOut The application should not pass on sensitive data in plain text to third-party software. The use of third-party software for securing a communication channel or a local storage container is permitted.

OSP.Purpose Any data collection, processing, storage, forwarding and deletion may only take place with a purpose limitation. The developer shall publish the lawful purpose of the application and, in addition, which data is processed and how, where and for how long it is stored. The permissible communication behavior and the internal and external sensors used must be selected based on the legitimate purpose.

Application note: Location data such as WiFi-SSID, GPS etc. may only be processed if it is essential for the functionality of the application. The data collected this way may only be processed for a specific purpose. It may not be persisted directly or indirectly (e.g. in image recordings) in the device unless it is directly required by the processing purpose.

OSP.RNG Random numbers must be obtained from a random number generator with high entropy. The application should initially introduce entropy from the user into the platform's random number generator once. The application then obtains randomness from the backend system and feeds it into the local random number generator.

OSP.SecurityLifeCycle The manufacturer shall implement a development cycle, the sub-steps of which are designed to strengthen the safety of the application. This includes measures to detect malicious activities and to initiate appropriate countermeasures by the operator.

[4] Third-party software is understood to be the sum of functionalities that were not developed by the developer of the application and that are not part of the functionality of the operating system platform used.

OSP.SecurityLog	If the app is connected to a backend system, logs of security incidents are sent to the backend system.

2.4.4 Residual risks

The operation of applications in the healthcare sector has particularly high requirements that cannot be adequately covered by existing devices and cloud solutions. The Technical Guideline therefore points out existing residual risks.

Mobile devices are particularly susceptible to theft. Even when using secure sources, it cannot be ruled out that malware is offered for download on these sources. Installed malware can exploit existing vulnerabilities.

Operating the backend system with public cloud providers involves particular risks for users' sensitive data. While high entropy, secure communication and encryption procedures mitigate risks, data in the cloud is potentially unprotected during processing. This places high demands on the cloud operator, as well as on other users who may be using resources on the same physical machine at the same time. By overcoming separation mechanisms, an attacker gains access outside their client area and may be able to view and manipulate sensitive data from another client (in this case: the healthcare applications) while it is being processed.

Communication connections between the platform, the application and the backend system are protected using the cryptographically secured TLS protocol. In this scenario, the TR assumes one-sided authentication, whereby the application checks the authenticity of the backend system. The application inserts its own randomness into the process of establishing the TLS connection in order to make it more difficult for an attacker to penetrate the TLS connection. However, random numbers on smartphone platforms generally do not achieve the quality required to protect sensitive data within a healthcare application. The residual risk during connection establishment is that the attacker can fake the authenticity of their own messages. This could allow the attacker to view and manipulate sensitive data transmitted from the application to the backend system.

In general, based on the information presented in chapter 2.1 and 2.3 in relation to the scope of the Technical Guideline, it is not possible to make an overall statement on the safety of the application, even taking into account all the test aspects listed. In order to increase the safety of the entire application, it is necessary to study further literature. This applies in particular to protection against attacks that directly target the backend system used and when connecting digital health applications with IoT devices.

3 Objectives for Healthcare Applications

3.1 Objectives

Testing in accordance with the Technical Guideline covers the minimum-security features of applications on mobile devices. The security functionality to be tested can be divided into the following objectives:

(1) Intended use

(2) Architecture

(3) Source code

(4) Third-party software

(5) Cryptography

(6) Authentication

(7) Data security

(8) Paid resources

(9) Network communication

(10) Platform-specific interactions

(11) Resilience

In case the functionality to be protected is used by the application, the manufacturer must document for each objective how its requirements are ensured.

3.1.1 Objective (1): Intended use

O.Purp_1	The developer MUST disclose the lawful purposes of the application and the processing of personal data prior to installation (e.g. in the description of the app store; cf. Appendix A) and inform the user of this at least when the application is first used.
O.Purp_2	The application MUST NOT collect and process data that does not serve the legitimate purpose of the application.
O.Purp_3	The application MUST obtain an active and unambiguous declaration of consent from the user prior to any collection or processing of personal data.
O.Purp_4	Data that the user has not expressly consented to be processed MUST NOT be collected, received or used by the application or the backend system.
O.Purp_5	The application MUST allow the user to withdraw consent that has already been given. The user MUST be informed about the possibility of withdrawal and the resulting changes in the behavior of the application before consent is given.
O.Purp_6	The developer MUST maintain a directory that shows which user consents have been given. The user-specific part of the directory MUST be automatically accessible to the user. It SHOULD be possible to request a history of this directory.
O.Purp_7	If the application uses third-party software, all functions used MUST be necessary for the legitimate purposes of the application. The application SHOULD safely disable other functions. If only a single or very few functions of the third-party software are required, it MUST be balanced whether the inclusion of all the third-party software is proportionate to the increase in the attack surface caused by the third-party software used.

O.Purp_8 Unless it is necessary for the primary or legitimate purpose of an application, sensitive data MUST NOT be shared with third parties. This includes storing data in parts of the file system to which other applications have access. The application MUST fully inform the user of the consequences of any sharing of application data that serves the primary or legitimate purpose and obtain the user's consent (OPT-IN).

O.Purp_9 The application MUST NOT display sensitive data on the screen unless this is necessary for the primary purpose of the application.

3.1.2 Objective (2): Architecture

O.Arch_1 Security MUST be an integral part of the software development and life cycle for the entire application (cf. “iOS Security Framework” [iOSSF] or “Design for Safety” [DfS]).

O.Arch_2 Already in the design phase of the application, it MUST be taken into account that the application will process sensitive data in the production phase. The architecture of the application MUST ensure the secure collection, processing, storage and deletion of sensitive data in a data lifecycle.

O.Arch_3 The lifecycle of cryptographic key material MUST follow an elaborate policy that includes properties such as the random number source, detailed key segregation of duties, key certificate expiration, integrity assurance through hashing algorithms, etc. The policy SHOULD be based on recognized standards such as [TR02102-2] and [NIST80057] should be used.

O.Arch_4 Sensitive data stored in backups MUST be encrypted according to the current state of the art.

O.Arch_5 Security functions MUST always be implemented on all external interfaces and API endpoints.

O.Arch_6 The application MUST enable the verification of integrity by means of a digital signature. The authenticity of the application is ensured by the trustworthiness of the source (see A.Source) is ensured.

O.Arch_7 If the application uses third-party software (e.g. for object serialization), the developer MUST ensure that only such third-party software is used whose functions can be used safely and that information about the scope of use and the security mechanisms used is clearly presented to the user. The application MUST use these functions securely. The developer MUST also ensure that unused functions cannot be activated by third parties.

O.Arch_8 Interpreted code[5] that interacts with user input (e.g. WebViews with JavaScript) MUST NOT have access to encrypted storage or user data unless it is absolutely necessary for the fulfilment of the primary purpose of the application.

O.Arch_9 The manufacturer MUST provide the user with a low-barrier way to report security issues. Communication SHOULD take place via an encrypted channel.

O.Arch_10 The application SHOULD check for available security-related updates at startup. If a security-relevant update is available, the application MUST NOT process sensitive data without installing this update. The user MUST be informed about the possibility of an update and about an update that has been carried out.

O.Arch_11 The developer MUST choose a source for publishing the application (and its updates) that protects the application against tampering by unauthorized parties and provides a trusted channel for obtaining the application.

[5] This does not refer to code of platform-specific programming languages.

O.Arch_12 The manufacturer MUST provide the app user with easily and securely accessible ways to obtain the application (e.g. in the form of a link list on its website, as an individually delivered QR code, etc.).

3.1.3 Objective (3): Source code

O.Source_1 The application MUST check all inputs before processing them in order to filter out potentially malicious values before processing.

O.Source_2 The application MUST mask incoming and outgoing data or clean it of potentially malicious characters or refuse to process it.

O.Source_3 Error messages and log files MUST NOT contain sensitive data (e.g. user identifiers).

O.Source_4 Potential exceptions in the program flow MUST be caught, handled in a controlled manner and documented. Technical error descriptions (e.g. stack traces) MUST NOT be displayed to the user.

O.Source_5 In the event of exceptions during program execution, the application SHOULD cancel access to sensitive data and securely delete it from memory.

O.Source_6 In program environments with manual memory management (i.e. the application itself can specify exactly when and where memory is read and written), the application MUST use secure function alternatives (e.g. printf_s instead of printf) for read and write access to memory segments.

O.Source_7 The application MUST ensure**Fehler! Textmarke nicht definiert.** that all sensitive data is securely deleted immediately after its processing purpose has been fulfilled.

O.Source_8 All options to support development (e.g. developer URLs, test methods, remnants of debug mechanisms, etc.) MUST be completely removed in production.

O.Source_9 Modern security mechanisms such as obfuscation and stack protection SHOULD be activated to build the application.

O.Source_10 Tools for static code analysis SHOULD be used for the development of the application.

3.1.4 Objective (4): Third-party software

O.TrdP_1 The provider[6] MUST maintain a centralized and complete list of dependencies on third-party software.

O.TrdP_2 Third-party software MUST be used in the latest version or the previous version intended for publication.

O.TrdP_3 Third-party software MUST be regularly checked for vulnerabilities by the developer (by evaluating publicly available information or by static/dynamic test methods). Remnants of options to support development (cf. O.Source_8) are to be considered a vulnerability. For all publicly known vulnerabilities, the manufacturer MUST analyze the extent to which the vulnerability affects the security of the overall system. Software or functions from third-party software MUST NOT be used for known vulnerabilities that affect the security of the overall system.

O.TrdP_4 Security updates for third-party software MUST be integrated promptly and made available to the user via an update. The manufacturer MUST submit a security concept that defines the tolerated continued use for the application or the backend system based

[6] Provider describes the legal entity responsible for the content of the product.

on the criticality of exploitable vulnerabilities. After the grace period has expired, the application MUST refuse operation.

O.TrdP_5 Before using third-party software, its source MUST be checked for trustworthiness.

O.TrdP_6 The application SHOULD not pass on sensitive data to third-party software.

O.TrdP_7 Data received via third-party software MUST be validated.

O.TrdP_8 Third party software that is no longer maintained MUST NOT be used.

3.1.5 Objective (5): Cryptography

O.Cryp_1 When using encryption in the application, permanently programmed secret or private keys MUST NOT be used.

O.Cryp_2 The application MUST rely on proven implementations for the realization of cryptographic primitives and protocols (cf. [TR02102-2]).

O.Cryp_3 The choice of cryptographic primitives MUST be appropriate to the use case and reflect the current state of the art (see [TR02102-1]).

O.Cryp_4 Cryptographic keys MUST NOT be used for more than exactly one purpose.

O.Cryp_5 The strength of the cryptographic keys MUST correspond to the current state of the art (see [TR02102-1]).

O.Cryp_6 All cryptographic keys SHOULD be stored in an environment protected against manipulation and disclosure.

O.Cryp_7 All cryptographic operations SHOULD take place in an environment protected from manipulation and disclosure.

3.1.5.1 Random numbers

O.Rand_1 All random values MUST be generated by a strong cryptographic random number generator which has been seeded with sufficient entropy (cf. [TR02102-1]).

3.1.6 Objective (6): Authentication

O.Auth_1 The manufacturer MUST provide a concept for authentication at an appropriate level of trust (cf. [TR03107-1]), for authorization (role concept) and for terminating an application session.

O.Auth_2 The application SHOULD implement authentication mechanisms and authorization functions separately. If different roles are required for the application, authorization MUST be implemented separately for each data access.

O.Auth_3 Each authentication process of the user MUST be implemented in the form of two-factor authentication.

O.Auth_4 In addition to the information specified in O.Auth_1 defined authentication at an appropriate level of trust, the manufacturer MAY offer the user an authentication option at a lower level of trust in accordance with Section 139e (10) SGB V, following comprehensive information and consent. This includes offering additional procedures based on the digital identities in the healthcare sector in accordance with Section 291 (8) SGB V.

O.Auth_5 Additional information (e.g. the end device used, the WiFi access node used or the time of access) SHOULD be included in the evaluation of an authentication process.

O.Auth_6	The user SHOULD be given the option of being informed about unusual login processes.
O.Auth_7	The application MUST implement measures to make it more difficult to try out login parameters (e.g. passwords).
O.Auth_8	If the application was interrupted (put into backend operation), a new authentication MUST be carried out after an appropriate period (grace period) has expired.
O.Auth_9	The application MUST request re-authentication after an appropriate period of inactivity (idle time).
O.Auth_10	The application MUST request re-authentication to reactivate the server session after an appropriate period of active use (active time).
O.Auth_11	The authentication data MUST NOT be changed without re-authenticating the user.
O.Auth_12	The application MUST use state-of-the-art authentication for the connection of a backend system.
O.Auth_13	Authentication data, such as session identifiers or authentication tokens, MUST be protected as sensitive data.
O.Auth_14	The application MUST allow the user to invalidate one or all previously issued session identifiers or authentication tokens.
O.Auth_15	If an application session is properly terminated, the application MUST inform the backend system so that session identifiers or authentication tokens are securely deleted. This applies to active termination by the user (log-out) as well as to automatic termination by the application (cf. O.Auth_9 and O.Auth_10).

3.1.6.1 Passwords

O.Pass_1	Strong password guidelines MUST exist for authentication using a user name and password. These SHOULD be based on current best practices.
O.Pass_2	To set up authentication using username and password, the strength of the password used MAY be displayed to the user. Information about the strength of the chosen password MUST NOT be saved.
O.Pass_3	The user MUST have the option to change their password.
O.Pass_4	If the application uses a backend system, the changing and resetting of passwords MUST be logged. The backend system SHOULD inform the user about changing and resetting passwords. To do so, a separate channel MUST be used.
O.Pass_5	If passwords are stored, they MUST be hashed using a hash function that complies with current security standards and using suitable salts.

3.1.7 Objective (7): Data security

O.Data_1	The factory setting of the application MUST provide maximum security.
O.Data_2	The application MUST store sensitive data in encrypted form. The operating systems own file system encryption is not sufficient. The key material for this encryption MUST NOT be persisted unencrypted. This applies to both volatile storage (e.g. in RAM) and permanent storage (e.g. in a cloud environment). Hardware-supported key management of the platform SHOULD be preferred.
O.Data_3	The application SHOULD store sensitive data in an area specially protected against access and manipulation.

O.Data_4	The application MUST NOT make resources that allow access to sensitive data available to third parties.
O.Data_5	All sensitive data collected MUST NOT be kept in the application beyond the duration of their respective processing.
O.Data_6	The application MUST comply with the principles of data minimization and purpose limitation.
O.Data_7	If the application is connected to a backend system, sensitive data SHOULD be stored and processed in the backend system.
O.Data_8	When using recording devices (e.g. camera), all metadata with data protection relevance, such as GPS coordinates of the recording location, hardware used, etc. MUST be removed.
O.Data_9	When collecting sensitive data through the use of recording devices (e.g. camera), it MUST be prevented that other applications gain access to it.
O.Data_10	When entering sensitive data via the keyboard, the application SHOULD prevent recordings from becoming recognizable to third parties.
O.Data_11	When entering sensitive data, the export to the clipboard SHOULD be prevented. Alternatively, the application MAY implement its own clipboard, which is protected from access by other applications.
O.Data_12	Sensitive data such as biometric data or private keys MUST NOT be exported from the component on which they were generated.
O.Data_13	The application MUST use functions of the operating system to prevent the display of sensitive data and access for third parties. This also includes the storage of the screen (e.g. screenshots and displays for app switching).
O.Data_14	The application MUST ensure**Fehler! Textmarke nicht definiert.** that all sensitive data is encrypted when the mobile device is locked.
O.Data_15	The application MUST encrypt locally stored data with a secure device binding.
O.Data_16	The developer MUST ensure that all sensitive data and application-specific login information on the mobile device is no longer accessible when the application is uninstalled.
O.Data_17	The application MUST allow the user to completely delete or make inaccessible all sensitive data and application-specific credentials.
O.Data_18	To counteract the misuse of sensitive data after a device loss, the application MAY implement a kill switch, i.e. a deliberate, secure overwriting of user data in the device at application level, triggered by the backend system. The manufacturer MUST protect the triggering of the kill switch by the user via the backend system against misuse by means of strong authentication mechanisms.

3.1.8 Objective (8): Paid resources

O.Paid_1	The application MUST make it clear to the user which paid services (e.g. additional functionalities or premium access) and which paid resources (e.g. SMS, phone calls, mobile data) are offered or used by the application.
O.Paid_2	The application MUST obtain the user's consent before using paid services.
O.Paid_3	The application MUST obtain the user's consent before requesting access (e.g. Android permissions) to paid resources.

O.Paid_4	The application MAY obtain the user's permanent consent for access to frequently used paid resources or paid services.
O.Paid_5	The application MUST enable the user to withdraw previously given consent.
O.Paid_6	The application SHOULD store the transaction history of paid services in the backend system. The transaction history, including metadata, MUST be stored as sensitive data according to O.Purp_8.
O.Paid_7	If the application offers paid services, the manufacturer MUST present a concept that prevents third parties from being able to trace the payment flows for the use of application functions.
O.Paid_8	The application MUST provide the user with an overview of the costs incurred. If the costs were incurred due to individual accesses, the application MUST provide an overview of the accesses.
O.Paid_9	The validation of completed payment transactions SHOULD be carried out in the backend system.
O.Paid_10	Third-party payment methods MUST meet the requirements for third-party software (see section 3.1.4).

3.1.9 Objective (9): Network communication

O.Ntwk_1	All network communication of the application MUST be encrypted end-to-end with mutual authentication (e.g. using mTLS).
O.Ntwk_2	The configuration of the encrypted connections MUST comply with the current state of the art (cf. [TR02102-2]).
O.Ntwk_3	The application MUST use security-checked third-party software to establish secure communication channels. This includes the library supplied with the operating system.
O.Ntwk_4	The application MUST use certificate pinning.
O.Ntwk_5	The application MUST check the server certificate of the backend system.
O.Ntwk_6	The application MUST validate the integrity and authenticity of responses from the backend system.
O.Ntwk_7	Platform-specific options such as “Cleartext Traffic Opt-out” and “In-App Transport Security” MUST be used.
O.Ntwk_8	The application MUST keep log files for all connections. These SHOULD be transmitted to the background system.

3.1.10 Objective (10): Platform-specific interactions

O.Plat_1	To use the application, the end device SHOULD have activated device protection (password, pattern lock, etc.). If device protection is not activated, the manufacturer MUST inform the user of the associated risks.
O.Plat_2	The application MUST NOT request permissions that are not necessary for the fulfillment of its primary purpose.
O.Plat_3	The application MUST inform the user of the purpose of the permissions to be requested and the consequences if the user does not grant them.
O.Plat_4	The application MUST NOT include sensitive data in messages or notifications that have not been explicitly enabled by the user (see O.Plat_5).

O.Plat_5	The application MAY offer the option to display messages and notifications to the user, including those containing sensitive data. This MUST be deactivated by default.
O.Plat_6	The application MUST implement access restrictions to sensitive data.
O.Plat_7	The platform-specific mechanisms for inter-process communication MUST NOT be used to exchange sensitive data unless they are necessary to fulfill the primary purpose of the application.
O.Plat_8	If the application uses a rendering engine to fulfill its primary purpose, it MUST configure it so that active content is not executed. If active content is essential for the realization of the application, the application MUST restrict the reloading of content to sources that are under the control of the developer or authorized by it.
O.Plat_9	If the application switches to background mode, it MUST remove all sensitive data from the current view ("Views" in iOS or "Activities" in Android).
O.Plat_10	The application MUST disable all unneeded protocol handlers in rendering engines.
O.Plat_11	The application MUST request the deletion of all application-specific session data (e.g. cookies) from the rendering engine used before it is properly terminated.
O.Plat_12	The application SHOULD have safely overwritten all user-specific data in the working memory after termination.
O.Plat_13	The user MUST be informed about security measures if they can be implemented by the user.
O.Plat_14	An aborted start[7] SHOULD be logged as a security event. The logging SHOULD take place in the backend system.

3.1.11 Objective (11): Resilience

O.Resi_1	The application MUST provide the user with accessible best practice recommendations for the safe use of the application and its configuration.
O.Resi_2	The application MUST use the operating system function to query whether the operating system is in an operating state that meets the requirements of the operating system manufacturer. If the operating system is not in a state specified by the operating system manufacturer, the application MUST respond appropriately. The application MUST show the user what risks exist for the user's data if the application is continued (e.g. that these could be disclosed) or prevent the continuation.
O.Resi_3	The application MUST implement its own checking mechanisms to determine whether it is running in a development/debug environment when the application is started. If the application detects that it is running in a development/debug environment, it MUST exit immediately.
O.Resi_4	The Application MUST implement its own checking mechanisms to determine whether the application has been started under unusual user rights. If the application detects that this is the case, it MUST exit immediately.
O.Resi_5	The application SHOULD ensure[8] the integrity of the end device before processing sensitive data.

[7] An aborted start is any interruption when opening the application. This also includes bringing the application back to the foreground (waking it up).

[8] Ensuring means querying a property or a status and then checking the query for a positive result.

O.Resi_6	The application MUST check the authenticity of the background system before accessing it (see also O.Ntwk_4).
O.Resi_7	The application SHOULD implement hardening measures, such as an integrity check before each processing of sensitive data within the program flow.
O.Resi_8	The application MUST implement state of the art measures against reverse engineering.
O.Resi_9	The application MUST take into account different versions of a platform (e.g. Android) from different manufacturers. It must be built in such a way that such different platforms or different platform versions do not lead to misbehavior. In particular, improper access to resources by different versions of a platform MUST be excluded.
O.Resi_10	The application MUST be robust against interference.

4 Audit steps of an application in the healthcare sector

4.1 Audit requirements

The audit of applications in the healthcare sector is based on the objectives described in chapter 3.1. Chapter 4.3 derives test characteristics from the objectives, which extend the requirements by an audit depth and notes for the auditors. The developer can make supporting statements in which he outlines the relevant implementation and provides a reference to the respective implementation. In case of complex audit characteristics, the manufacturer provides a comprehensive list of occurrences. Depending on the depth of auditing implemented, these developer statements support the audit. The table below shows which steps are required as a minimum for the respective audit depth.

Table 2: Audit depths and minimum requirements

Audit depth	Minimum requirements for verification
CHECK	The auditor checks (analogue to the use of terms in Common Criteria Evaluation Methodology) the measure described by the manufacturer with regard to its effectiveness and dispels any doubts (plausibility check) as to whether the test aspect and the associated security problem are comprehensively addressed by the measures described. In doing so, the auditor MUST take into account the current state of the art for the respective platform. The validation MAY include further steps, such as a source code analysis, if the auditor requires these for a comprehensive assessment.
EXAMINE	The auditor examines (analogue to the use of terms in Common Criteria Evaluation Methodology) the test characteristic in question. The auditor MUST go beyond the minimum requirements for "CHECK" in his examination: As a rule, this will be done through comprehensive source code analysis of the relevant implementation parts and penetration testing. Support from the manufacturer can be used. "EXAMINE" always requires an independent assessment by the auditor.

The use of a source text analysis during the assessment also follows from the audit depths. For "CHECK", the TR reviewer selects how high the coverage of the analysis is necessary for his assessment. For "EXAMINE", the Technical Guideline reviewer must explain the extent to which all relevant lines of code were taken into consideration.

4.2 Recording the results

The audit results must be recorded in a way that enables uninvolved third parties to repeat the audit steps based on the information in the report and achieve the same result. For this purpose, in addition to the description of the individual audit steps, it is necessary that the tools used are visible in the report. The table below defines the permissible results that can result from auditing a characteristic. The auditor explains how he arrived at a corresponding result.

Table 3: Possible test results

Result	Necessary information
PASS	The auditor explains his understanding of why the manufacturer's implementation fulfills the required safety objective. The report details the test steps performed and the audit result.

Result	Necessary information
INCONCLUSIVE	The test report specifies/references the missing or inconsistent information so that the manufacturer can rectify the non-conformity to the relevant aspect of the safety objective.
FAIL	The audited application fails to meet the relevant security objective. The auditor documents the extent to which attacks can be prevented by security measures in the application's environment (e.g. operational measures). The auditor includes evidence of the violation of the test characteristic in the log. The tester includes the risk arising from the violation of the test characteristic in the risk assessment.
NOT APPLICABLE (N/A)	The tested application does not have any implementation of the functionalities to be protected by the test aspect. Therefore, the relevant test characteristic cannot be applied to the application to be tested.

The auditors identify existing residual risks when using the application in the healthcare sector. The identification of existing residual risks takes into account the fact that the loss of health data immediately leads to damage for the user and that sufficient protective measures could not be identified at the time of the TR test. The risk assessment must include at least the following aspects:

- Identification of risks from the failure to implement or inadequate implementation of "SHOULD" requirements in security targets.
- Implementation-specific risks.
- Risks due to integration in the planned operating environment.
- Consider the suitability of the monitoring and the response options provided in the product for the operator during product testing.

4.3 Audit characteristics

The audit characteristics extend the objectives from chapter 3 by their audit depth and supplementary information for auditor. The auditor should go beyond the individual audit steps to ensure that the security target in question is met overall. This may include additional audit characteristics not listed here.

4.3.1 Audit characteristics for objective (1): Intended use

Table 4: Audit characteristic: Intended use

Objective	Short version of the objective	Audit depth	Comments
O.Purp_1	Developer’s obligation to provide information on the lawful purpose and processing of personal data.	CHECK	The auditor checks whether a description exists and whether it corresponds to the lawful purposes of the application. The lawful purposes defined by the developer are used as the basis for this. A legal check of the legality is not required.

Objective	Short version of the objective	Audit depth	Comments
O.Purp_2	Earmarked collection and processing of data.	CHECK	The auditor uses the permission policy to check which data is processed in the application and whether it corresponds to the legitimate purposes of the application. The use of sensor data is only permitted to the extent that it is used to collect the seed, for example. The lawful purposes defined by the manufacturer are used as the basis for this. A legal check of the lawfulness is not required.
O.Purp_3	Obtain a declaration of consent from the user.	CHECK	The auditor checks whether personal data can be processed without the user's consent.
O.Purp_4	Use of consented data only.	CHECK	The auditor matches the values specified in O.Purp_2 with the approvals granted.
O.Purp_5	Enable withdrawal of consent.	CHECK	The auditor checks whether the user is given the opportunity to withdraw consent. In addition, the auditor validates that the user is informed of the consequences of withdrawing consent.
O.Purp_6	Maintain a directory of user consents.	CHECK	The auditor checks the existence, timeliness and completeness of the list.
O.Purp_7	Use only required third-party software.	CHECK	The auditor checks the developer's considerations for functions that do not serve the legitimate purpose of the application. For example, an API for social networks may only be used if this is compatible with the legitimate purpose of the application. The risk assessment covers the impact on the protection of health data, for example in the case of usage behavior in logging frameworks that can be identified by third parties.

Objective	Short version of the objective	Audit depth	Comments
O.Purp_8	Disclosure of sensitive data only for the primary or lawful purpose.	CHECK	The auditor checks the developer's considerations as to whether the disclosure of sensitive data to third parties serves the primary or lawful purpose of the application. In addition, it checks whether the transfer must always be explicitly permitted by the user (opt-in). Disclosure to services whose primary purpose is the processing of data for advertising purposes is generally prohibited. The risk assessment takes into account how the disclosure of data to third parties relates to the need for protection of the information (data) forwarded and the resulting risk of disclosure of information.
O.Purp_9	Only display sensitive data on the screen for a specific purpose.	CHECK	The auditor examines the developers' considerations as to whether the display of sensitive data is necessary at the given time to fulfill the purpose of the application. The risk assessment must take into account how the application protects the user from displaying sensitive data (cf. T.VisibleAsset).

4.3.2 Audit characteristics for objective (2): Architecture

Table 5: Audit characteristic: Architecture

Objective	Short version of the objective	Audit depth	Comments
O.Arch_1	Security is part of the software development and life cycle.	CHECK	The auditor checks whether the source code and the design documents indicate the use of current best practices during development.
O.Arch_2	Consideration of the processing of sensitive data in the design phase.	CHECK	The auditor checks design and architecture documents for consideration of the processing of sensitive data, including the data life cycle.

Objective	Short version of the objective	Audit depth	Comments
O.Arch_3	Documentation of the life cycle of cryptographic material.	CHECK	The auditor assesses the prepared guideline of the developer and its consideration in the risk assessment.
O.Arch_4	No unencrypted sensitive data in backups.	EXAMINE	The auditor uses source code analysis and practical tests to examine whether sensitive data is present in unencrypted backups.
O.Arch_5	Distributed implementation of security features.	EXAMINE	The auditor examines the presence and quality of security functions through source code analysis and practical tests. Security functions include authentication, authorization, input validation and the use of escape syntaxes.
O.Arch_6	The authenticity and integrity protection of the application.	EXAMINE	The auditor examines the existence and quality of the authenticity and integrity protection used through source code analysis and practical tests. In doing so, it is important that it is up to date (see [TR02102-1]) of the signature procedures used. The effectiveness against manipulation of the application (see T.MemoryStructures) must be considered in the risk assessment.
O.Arch_7	Secure use of third-party software functions.	EXAMINE	The auditor examines, through source code analysis and practical tests, that functionalities are used safely and unused functionalities are not accessible. It also checks whether the user is sufficiently informed about the use of third-party software.
O.Arch_8	Dedicated access to encrypted storage or user data through interpreted code.	EXAMINE	The auditor examines through source code analysis and practical tests whether access to encrypted memory or user data is possible via interpreted code. If this is the case, the auditor checks the developer's considerations regarding the compelling necessity for the fulfillment of the primary purpose and consideration in the risk assessment.

Objective	Short version of the objective	Audit depth	Comments
O.Arch_9	Low-barrier option to report security problems.	CHECK	The auditor checks whether a corresponding option is available. If no encrypted channel is provided, this must be taken into account in the risk assessment.
O.Arch_10	Forced updates from the backend system.	EXAMINE	The auditor examines the quality of the implementation of the corresponding functionality in the application. If this is not available, the auditor checks the developer's considerations regarding the effects on the security of the application. This must be taken into account in the risk assessment. If the functionality is available, the auditor uses practical tests to check whether blocking individual requests effectively prevents further use of the application.
O.Arch_11	Provision of updates via a dedicated app store.	CHECK	The auditor checks whether the developer provides its own app store. Resulting considerations and effects on security must be taken into account in the risk assessment.
O.Arch_12	Use of cryptographic measures for alternative download sources or mechanisms.	CHECK	The auditor checks the existence and quality of the procedures used via source text analysis.

4.3.3 Audit characteristics for objective (3): Source code

Table 6: Audit characteristic: Source code

Objective	Short version of the objective	Audit depth	Comments
O.Source_1	Checking inputs before processing.	CHECK	The auditor checks whether all inputs from untrusted sources have security functions according to O.Arch_5 are present. Input means any type of data that flows into the application. These are, for example, user inputs, inputs from third party components, etc.

Objective	Short version of the objective	Audit depth	Comments
O.Source_2	Use of an escape syntax in structured data.	CHECK	The auditor checks whether there is an escape syntax of structured data is available for all inputs according to O.Arch_5. Harmful signs must be considered according to context. In the database context, for example, quotation marks or percentage signs may be malicious, whereas in the web/HTML context, tag brackets (<) are more likely to be malicious. Therefore input validation must always be context-related. If a potentially malicious input is detected, it must be either masked or rejected. Rejecting should be preferred to masking. If masked inputs are passed on, they must be masked in a way that they do not have any harmful effects in the context in which they are passed on.
O.Source_3	No sensitive data in messages.	CHECK	The auditor checks whether sensitive data can be viewed via error messages or notifications.
O.Source_4	Controlled handling and documentation of exemptions.	EXAMINE	The auditor examines the controlled handling and documentation of exceptions through source text analysis and practical tests.
O.Source_5	Cancellation of access to sensitive data in exceptions.	EXAMINE	The auditor examines access to sensitive data in the event of exceptions in the program process. Any identified access must be considered in the risk assessment.
O.Source_6	Use of secure functional alternatives to access memory.	EXAMINE	The auditor uses source code analysis to examine if the application uses insecure access features. The test shall include all the source code implemented by the developer. Third party software is covered in the O.TrdP objectives.

Objective	Short version of the objective	Audit depth	Comments
O.Source_7	Secure deletion of sensitive data after processing.	EXAMINE	The auditor uses source code analysis and practical tests to examine whether all sensitive data not covered by O.Data_2 is securely deleted after processing. “Secure deletion” requires overwriting the data in memory. Attention must also be paid to any copies of the data. For programming languages without manual memory management, this includes replacing strings with byte arrays.
O.Source_8	Complete removal of supportive development options and debug mechanisms in the productive version.	EXAMINE	The auditor examines the productive version of the application for residues of options to support development and residues of character chains, debug mechanisms and debug information.
O.Source_9	Activating modern security mechanisms of the development environment.	CHECK	The auditor checks whether modern security mechanisms of the development environments have been used. If such security mechanisms cannot be implemented, this must be considered in the risk assessment.
O.Source_10	Use of tools for static code analysis.	CHECK	The auditor checks by analyzing the source code whether tools for static code analysis were used during the development. If no tools were used, this has to be considered in the risk assessment.

4.3.4 Audit characteristics for objective (4): Third-party software

Table 7: Audit characteristic: Third-party software

Objective	Short version of the objective	Audit depth	Comments
O.TrdP_1	Dependencies from third party software.	CHECK	The developer provides a list of third-party software, including the versions used. The auditor checks the list provided for completeness.

Objective	Short version of the objective	Audit depth	Comments
O.TrdP_2	Use of current version in third-party software.	CHECK	The auditor checks the entries listed in O.TrdP_1 to ensure that the versions of the third-party software used are up to date. The considerations regarding the selected versions are taken into account in the risk assessment.
O.TrdP_3	Developer checks the third-party software for vulnerabilities.	CHECK	The developer provides an overview of the latest vulnerability analysis of the third-party software used. This overview is checked by the auditor and taken into account in the risk assessment. In addition, the auditor checks whether the manufacturer provides a mitigation strategy within an appropriate grace period if vulnerabilities occur.
O.TrdP_4	Security concept for installation of security updates for third-party software.	CHECK	The auditor checks the existence of such a concept. A content check is not required within this audit. In addition, the auditor checks whether the manufacturer provides a mitigation strategy.
O.TrdP_5	Trustworthiness of the source of third-party software.	CHECK	The auditor checks the manufacturer's measures for verifying the trustworthiness of third-party providers.
O.TrdP_6	No transfer of sensitive data to third party software.	EXAMINE	The auditor examines through source code analysis and practical tests that no sensitive data is passed on to third-party software. An exception to this is the transfer of data that is required for the primary or legitimate purpose of the application (e.g. third-party software for transport encryption). Risks resulting from non-compliance must be taken into account in the risk assessment.
O.TrdP_7	Validation of incoming data via third-party software.	CHECK	The auditor checks whether incoming data via third-party software are handled in accordance with O.Source_1 and security functions are available in accordance with O.Arch_5.

Objective	Short version of the objective	Audit depth	Comments
O.TrdP_8	Checking the maintenance of third-party software used.	CHECK	The auditor checks whether the third-party software used is actively maintained by the developer. Software is considered to be no longer maintained if security-critical vulnerabilities are known but have not been repaired within a reasonable period of time.

4.3.5 Audit characteristics for objective (5): Cryptography

Table 8: Audit characteristic: Cryptography

Objective	Short version of the objective	Audit depth	Comments
O.Cryp_1	No hard-coded keys or other secrets.	EXAMINE	The auditor examines whether hard-coded secret or private keys are used. Excluded are techniques that strongly conceal the key used from reverse engineering according to the current state of the art (keyword: "white box cryptography"). If cascaded encryption is used, at least one encryption level should be strongly protected against reverse engineering and at least one non-static key should be used.
O.Cryp_2	Only proven implementations in cryptographic primitives.	EXAMINE	The auditor examines the list of crypto implementations used against the current state of the art (cf. [TR02102-2]).
O.Cryp_3	Appropriate choice of cryptographic primitives.	EXAMINE	The auditor examines the developer's considerations regarding the choice of cryptographic primitives and checks whether these correspond to the current state of the art (cf. [TR02102-1]).
O.Cryp_4	Purpose limitation of cryptographic keys.	EXAMINE	The auditor examines the cryptographic keys used for their purpose. A distinction is made between the purpose of protection through encryption and authentication.
O.Cryp_5	Use of strong cryptographic keys.	EXAMINE	The auditor tests the strength of the keys used against the current state of the art (see [TR02102-1]).

Objective	Short version of the objective	Audit depth	Comments
O.Cryp_6	Protection of cryptographic keys against manipulation by the environment.	EXAMINE	The auditor examines the environment for storing cryptographic keys. Secure Enclaves, embedded Secure Elements, Trusted Execution Environments etc. are considered secure environments. Operation on all approved hardware platforms must be taken into account. In the event of non-compliance, the auditor checks the developers' considerations regarding the impact on the security of the applications or discusses a lack of protection in the risk assessment.
O.Cryp_7	Manipulation protection of cryptographic operations by environment.	EXAMINE	The auditor checks the environment for the execution of cryptographic operations analogous to O.Cryp_6. Operation on all approved hardware platforms must be taken into account.
O.Rand_1	Generation of random values using a secure random number generator.	EXAMINE	The auditor examines the quality of the cryptographic random number generator through source code analysis and practical tests. Information on sufficiently secure random number generators can be found in [TR02102-1] Chapter 10. For the post-processing of the random numbers, the algorithms considered sufficiently secure by the BSI (see [AIS20], [TR03107-1] and [TR03116-4]) must be used.

4.3.6 Audit characteristics for objective (6): Authentication

Table 9: Audit characteristic: Authentication

Objective	Short version of the objective	Audit depth	Comments
O.Auth_1	Manufacturer concept for authentication, authorization and termination of application sessions.	CHECK	The auditor checks the concept provided by the developer for authentication, authorization and termination of the application session. It assesses the quality of the procedures used based on the current state of the art. According to the BSI's assessment, there are currently no consumer devices that can use biometrics for identification or authentication at a "high" level of assurance in an unmonitored application scenario.
O.Auth_2	Separate implementation of authentication and authorization.	EXAMINE	The auditor examines the measures taken to separate authorization and authentication mechanisms. If the mechanisms are not separated, the developers' considerations must be taken into account in the risk assessment.
O.Auth_3	Two-factor authentication.	EXAMINE	The auditor examines the existence and quality of the two-factor authentication through source code analysis and practical tests. In particular, it checks whether the factors used originate from different categories (knowledge and possession) and are compatible with the authentication method described in O.Auth_1.

Objective	Short version of the objective	Audit depth	Comments
O.Auth_4	Authentication via additional procedures corresponding to a lower security level.	EXAMINE	The auditor examines the existence of authentication options with a lower security level. If such procedures are offered, the auditor checks through source code analysis and practical tests whether they offer adequate security. Appropriate security requirements for low-threshold procedures can be found in the current version of gematik GmbH's “Spezifikation Sektoraler Identity Provider” [gemSpec_IDP_Sek]. The developers’ considerations regarding the provision of additional authentication options and the chosen implementation must be taken into account in the risk assessment.
O.Auth_5	Include additional information when evaluating the authentication process.	EXAMINE	The auditor examines the presence and quality of additional information for the evaluation of an authentication process. Such information can be implemented, for example, via the invalidation or deletion of keys when biometric system features are changed or a check for changes to biometric metadata. A check for conformity with data protection of the information collected is not required within the scope of this TR. If no additional information is used for the assessment, the auditor checks the manufacturer's considerations. These must be taken into account in the risk assessment.
O.Auth_6	Inform the user about unusual login attempts.	CHECK	The auditor checks whether the user has easy access to information on registration processes. If this is not the case, the developers’ considerations must be examined and taken into account in the risk assessment.

Objective	Short version of the objective	Audit depth	Comments
O.Auth_7	Prevent login parameters from being tried.	CHECK	The auditor validates that login parameters are prevented from being tried out. This can be achieved, for example, by delaying subsequent login attempts or by using so-called captchas.
O.Auth_8	Re-authentication in case of interrupted application.	CHECK	The auditor validates that re-authentication must take place after a period of time appropriate to the application, during which it was switched to background mode. The quality of the required authentication must be appropriate to the level of trust (cf. O.Auth_3).
O.Auth_9	Re-authentication after an appropriate period of time during which the application was not actively used.	CHECK	The auditor validates that re-authentication must take place after a period of time appropriate to the application during which it has not been actively used. The quality of the required authentication must be appropriate to the level of assurance (cf. O.Auth_3).
O.Auth_10	Re-authentication after an appropriate period of time during which the application has been in permanent active use.	CHECK	The auditor validates that re-authentication must take place after a period of time appropriate to the application during which it has been in continuous active use. The quality of the required authentication must be appropriate to the level of assurance (cf. O.Auth_3).
O.Auth_11	Sufficient authentication of the user to change the authentication data.	EXAMINE	The auditor examines whether it can change the authentication data without appropriate authentication. This also applies to a password reset procedure. If this process is based on security queries, for example, the answer must not be easy to guess or even be able to be determined from possibly public information (e.g. mother's maiden name).
O.Auth_12	Authentication at the interface between application and backend system.	CHECK	The auditor checks whether the application supports authentication of the backend system.

Objective	Short version of the objective	Audit depth	Comments
O.Auth_13	Protection of authentication data.	CHECK	The auditor checks whether authentication data is treated as sensitive data in accordance with the requirements of this TR.
O.Auth_14	Invalidation of authentication data by the user.	CHECK	The auditor checks whether the application allows the user to invalidate one or all previously issued authentication data.
O.Auth_15	Notification of the backend system about terminated application sessions by the application.	CHECK	The auditor checks whether the backend system is informed when the application session is properly terminated by the application.
O.Pass_1	Enforce strong password policies.	CHECK	The auditor checks whether password guidelines that correspond to the current state of the art are used. Otherwise, the developers' considerations must be checked and taken into account in the risk assessment.
O.Pass_2	Display the strength of the password used.	EXAMINE	The auditor examines whether the strength of the password used is displayed to the user. If this is the case, it uses source code analysis and practical tests to check whether information about the password or its quality remains in the application memory.
O.Pass_3	Option to change the password.	CHECK	The auditor checks whether the user has the option to change their password and verifies that this functionality cannot be misused.
O.Pass_4	Logging and information about changing and resetting of passwords.	CHECK	The auditor checks the existence and quality of additional information for logging changes and resets of passwords.

Objective	Short version of the objective	Audit depth	Comments
O.Pass_5	Use of cryptographically secure hashing algorithms and salts to store passwords.	EXAMINE	The auditor examines whether passwords are stored in the application. It verifies that the protection mechanisms used meet the current state of the art and the requirements for hash functions, number of iterations and salts (cf. [TR02102-1]). Measures that slow down brute force attacks are taken into account in the risk assessment.

4.3.7 Audit characteristics for objective (7): Data security

Table 10: Audit characteristic: Data security

Objective	Short version of the objective	Audit depth	Comments
O.Data_1	Maximum security with factory settings.	CHECK	The auditor checks the default settings of the application when it is installed. This includes, among other things, the operating system authorizations that the application requests. The authorizations of the mobile device must serve the purpose of the application and may only be requested as soon as they are used.

Objective	Short version of the objective	Audit depth	Comments
O.Data_2	Encrypt all sensitive data.	EXAMINE	The auditor validates that sensitive data (see Appendix A) can only be stored by the application in encrypted form. The auditor also examines whether hardware-supported key management of the operating system is used to store the necessary keys. If sufficient protection of the keys is ensured by the platform (e.g. in an embedded secure element/trusted execution environment), the application must effectively protect these keys against disclosure. The auditor includes the effectiveness against reverse engineering in the risk assessment.
O.Data_3	Storage of sensitive data.	EXAMINE	The auditor examines whether hardware-supported measures (such as Trusted Execution Environment) are used to store sensitive data. If this is not the case, the auditor includes the considerations of effectiveness versus reverse engineering in the risk assessment.
O.Data_4	Access to sensitive data by third parties.	EXAMINE	The auditor examines whether the application provides resources through which third parties can gain access to sensitive data. This includes data in shared storage areas, services or interfaces via which sensitive data is provided.
O.Data_5	Deletion of all sensitive data collected after processing by the application has been completed.	CHECK	The auditor checks whether data is kept in the application beyond the period of its processing. Data that is no longer used must be securely deleted.

Objective	Short version of the objective	Audit depth	Comments
O.Data_6	Collection, storage and processing of data required exclusively for the purpose of the application.	CHECK	The auditor checks which data is collected, stored and processed by the application and compares this with the purpose of the application.
O.Data_7	Storage and processing of sensitive data.	CHECK	The auditor checks which data the application permanently stores or processes. It determines the risk posed by such storage and processing in the application and includes it in the risk assessment.
O.Data_8	Removal of metadata with data protection relevance.	CHECK	The auditor checks whether the application can collect data that contains metadata. In this case, the auditor checks whether metadata with data protection relevance is removed before further processing, such as transfer to the backend system.
O.Data_9	Access restriction when collecting sensitive data.	EXAMINE	The auditor checks whether collected sensitive data is made available to other applications on the device or whether data is stored in public directories.
O.Data_10	No recordings when entering sensitive data via the keyboard.	EXAMINE	The auditor examines whether sensitive data can be entered via software keyboards of the operating system or third-party providers. This includes, in particular, caches, autocorrect and autocomplete procedures, third-party input devices and any storage that can be analyzed by third parties. If this is the case, the auditor checks the developers' considerations and takes them into account in the risk assessment.
O.Data_11	No export of sensitive data to the clipboard.	CHECK	If the application allows sensitive data to be exported to the operating system's clipboard, the outflow of this data must be taken into account in the risk assessment.

Objective	Short version of the objective	Audit depth	Comments
O.Data_12	No export of sensitive data from its origin.	EXAMINE	The auditor examines whether sensitive data that does not need to be exported can still be exported. This includes biometric data or private cryptographic keys.
O.Data_13	No storage of screen content or access by third parties when displaying sensitive data.	CHECK	The auditor checks whether the application prohibits the creation of screenshots. This includes both the active and passive creation of screenshots, such as those taken for the preview in the Task Manager. The auditor also checks whether sensitive data is not displayed in the application for longer than necessary. The residual risk of displaying sensitive data is taken into account in the risk assessment.
O.Data_14	Encryption of all sensitive data when the device is locked.	EXAMINE	Older mobile platform versions sometimes allow the application to be stored on external storage media that are not subject to storage encryption. The auditor checks whether the application prohibits such or comparable procedures.
O.Data_15	Device binding of locally stored data.	EXAMINE	The auditor examines whether data stored by the application can be processed on other devices. Data that is explicitly exported by the user for processing on other devices is excluded from this restriction.
O.Data_16	Remove or make all sensitive data inaccessible on the mobile device when uninstalling the application.	CHECK	The auditor checks whether data remains on the end device after the application has been uninstalled. If this is the case, the auditor continues to check whether this data contains sensitive information or allows conclusions to be drawn about sensitive data.

Objective	Short version of the objective	Audit depth	Comments
O.Data_17	Possibility to delete or make all sensitive data inaccessible of the application.	EXAMINE	The auditor validates that the user is given the option of deleting all sensitive data completely or making it inaccessible. In addition, he examines the effectiveness of the measures taken through practical tests.
O.Data_18	Secure overwriting of user data on the mobile device by the user via the backend system.	EXAMINE	The auditor deletes the data via the background system and validates that no more user data can be read on the mobile device.

4.3.8 Audit characteristics for objective (8): Paid resources

Table 11: Audit characteristic: Paid resources

Objective	Short version of the objective	Audit depth	Comments
O.Paid_1	Display of paid services and resources.	CHECK	The auditor validates that all chargeable services and resources are clearly recognizable as such.
O.Paid_2	User consent before carrying out paid services.	CHECK	The auditor validates that all chargeable services can only be provided after confirmation by the user.
O.Paid_3	User consent before requesting access to paid resources.	CHECK	The auditor validates that the use of services that may incur additional costs for the user (e.g. sending text messages) is only possible after the user has given their consent.
O.Paid_4	Permanent user consent to frequently used, paid services or resources.	CHECK	If the application requires the user's permanent consent to access paid resources, the auditor checks whether this is necessary for the primary purpose of the application (cf. O.Purp_1).
O.Paid_5	Allow withdrawal of consent.	CHECK	The auditor checks whether the application displays a list of all declarations of consent given by the user and whether these can be subsequently changed.

Objective	Short version of the objective	Audit depth	Comments
O.Paid_6	Storage of the sensitive transaction history in the backend system.	EXAMINE	The auditor uses practical tests and source code analysis to examine whether a transaction history is kept in the application. The transaction history should be stored securely in the backend system and be accessible from the application. If the transaction history is stored in the application itself, a risk assessment must show the extent to which the security of the stored data can be guaranteed.
O.Paid_7	Profiling by tracking cash flows by third parties.	CHECK	The auditor checks whether conclusions can be drawn about the characteristics or behavior of the user by tracking payment flows. The developers' considerations regarding potential conclusions must be taken into account in the risk assessment.
O.Paid_8	Display of the overview of costs incurred.	CHECK	The auditor checks whether the application provides the user with an overview of the costs incurred. If the costs were incurred due to individual accesses, the auditor checks whether the application provides an overview of the accesses.
O.Paid_9	Validation of payment transactions in the backend system.	EXAMINE	The auditor uses source code analysis and practical tests to examine whether the application can independently validate payments and, for example, activate functions that are subject to a charge.
O.Paid_10	Requirements for payment procedures of third party providers.	CHECK	The auditor checks the payment methods used by third-party providers. For both third-party software and web services, the auditor checks that no sensitive user data flows to the payment service provider (e.g. that the title of the booked service does not contain any sensitive information).

4.3.9 Audit characteristics for objective (9): Network communication

Table 12: Audit characteristic: Network communication

Objective	Short version of the objective	Audit depth	Comments
O.Ntwk_1	Network communication is encrypted throughout with mutual authentication.	EXAMINE	The auditor validates that encrypted communication between the application and other components is only possible with mutual authentication.
O.Ntwk_2	Configuration of the encrypted connection according to the current state of the art.	EXAMINE	The auditor validates that the data specified in O.Ntwk_1 corresponds to the state of the art (see [TR02102-2]) corresponds to the state of the art.
O.Ntwk_3	Secure communication channels only with operating system functions or security-checked third-party software.	EXAMINE	The auditor examines how a secure communication channel is established. If no operating system functions are used, the auditor validates that the third-party software used to terminate the connection fulfills the requirements described in chapter 3.1.4. Proprietary implementations for establishing secure communication channels are not permitted. According to A.OperatingSystem the functions for establishing secure communication channels in the operating system are assumed to be secure. Security-checked means that all security-relevant areas of the software have been checked for their security properties by another party.
O.Ntwk_4	Support for certificate-pinning.	EXAMINE	The auditor validates that the application supports certificate pinning and implements it effectively.
O.Ntwk_5	Validation of the certificate of the backend system.	EXAMINE	The auditor checks how the application validates the certificate of the background system.
O.Ntwk_6	Validation of the integrity and authenticity of the backend system responses.	EXAMINE	The auditor confirms that the integrity and authenticity of the messages of the backend system are validated by the application.

Objective	Short version of the objective	Audit depth	Comments
O.Ntwk_7	Use of platform-specific options.	EXAMINE	The auditor confirms that the platform-specific options for ensuring secure communication have been implemented.
O.Ntwk_8	Provision of complete log files for all established connections.	CHECK	The auditor checks the log files provided by the developer and validates that the HTTP headers are fully included. If no logging of security-relevant events takes place on the backend system, this aspect must be taken into account in the risk assessment.

4.3.10 Audit characteristics for objective (10): Platform-specific interactions

Table 13: Audit characteristic: Platform-specific interactions

Objective	Short version of the objective	Audit depth	Comments
O.Plat_1	Device protection required to use the application.	CHECK	The auditor checks whether the developer permits use without activated device protection. If this is the case, the auditor checks whether the user is adequately informed about the resulting risks and takes the developers' considerations into account in the risk assessment. According to A.OperatingSystem it is assumed in particular that the operating system has functions that enable the application to query the device protection setting.
O.Plat_2	Only request the authorizations required for the primary purpose.	CHECK	The auditor checks the authorizations required by the application and confirms that they are necessary for fulfilling the primary purpose of the application (O.Purp_1) are required.
O.Plat_3	Reference to the purpose of the authorizations and the consequences of not granting them.	CHECK	The auditor checks whether the application indicates the purpose of the required authorizations.

Objective	Short version of the objective	Audit depth	Comments
O.Plat_4	No sensitive data in messages or notifications.	EXAMINE	The auditor uses source code analysis and generated log messages to check whether the application writes sensitive data to these messages. If the logged data allows conclusions to be drawn about the user, this data outflow must be taken into account in the risk assessment.
O.Plat_5	Option to display messages/notifications containing sensitive data.	CHECK	If the application offers the option of displaying messages with sensitive data, the auditor checks whether these are deactivated by default. It also checks whether the user is adequately informed about the resulting risks when this option is activated. The developers' decision to offer such options must be taken into account in the risk assessment.
O.Plat_6	Access restrictions to sensitive data.	EXAMINE	The auditor examines whether the application restricts access to all data as long as it is under the control of the application. This includes restricting access to designated file paths to protect sensitive data through automatic saving or unintentional saving by the user. In addition, it must be validated that the application cannot send broadcast messages that can be read by all applications installed on the device. The application may only send messages to authorized applications. If the application does not implement this requirement, the developers' considerations must be included in the risk assessment. According to A.OperatingSystem it is assumed that the operating system has effective functions that isolate individual applications from each other and from the operating system. The isolation must be consistent and cover at least the process level, memory contents and file system level.

Objective	Short version of the objective	Audit depth	Comments
O.Plat_7	Use of sensitive functionalities via inter-process communication.	CHECK	If the application offers interfaces, the auditor takes their usability into account in the risk analysis. According to A.OperatingSystem it is assumed in particular that functions for inter-process communication are available for the operating system, for which rules can be set to isolate communication.
O.Plat_8	Use of rendering engines to reload active content.	CHECK	If the application uses rendering engines, the auditor checks whether the components prevent the reloading of active content or restrict it to sources under the manufacturer's control. The selection of permitted sources is taken into account in the risk analysis.
O.Plat_9	Removal of sensitive data when switching to background mode.	EXAMINE	The auditor checks whether the application removes sensitive data from the corresponding display elements of the application.
O.Plat_10	Deactivation of unneeded protocol handlers in rendering engines.	CHECK	If the application uses rendering engines, the auditor checks whether the components deactivate protocol handlers that are not required.
O.Plat_11	Delete application-specific session data when exiting the application.	EXAMINE	If the application uses rendering engines or other ways of displaying web content, the auditor checks whether application-specific session data is deleted after the application is closed.
O.Plat_12	Overwrite all user-specific data when closing the application.	EXAMINE	The application must not rely purely on the operating system and the runtime environment's garbage collector for termination. Sensitive data must be actively deleted or overwritten. The auditor determines the risks for the individual data concerned and takes these into account in the risk assessment.

Objective	Short version of the objective	Audit depth	Comments
O.Plat_13	Informing the user about necessary security measures for the application, third-party software and platforms.	CHECK	The auditor checks whether the user is informed and, if necessary, instructed about security measures that they can implement themselves. The auditor assesses whether the measures are sufficient to limit residual risks.
O.Plat_14	Logging of certain security events.	CHECK	The auditor checks the log files provided by the vendor and validates that an aborted start and other security events of the application are logged. The information is used for post-mortem analysis of security incidents and should therefore contain information about all outgoing connections, including meta-information about proxies used and server certificates checked.

4.3.11 Audit characteristics for objective (11): Resilience

Table 14: Audit characteristic: Resilience

Objective	Short version of the objective	Audit depth	Comments
O.Resi_1	Information on the safe handling of the product.	CHECK	The auditor checks whether the application provides “best practices”. It confirms that existing best practices correspond to the current state of the art.

Objective	Short version of the objective	Audit depth	Comments
O.Resi_2	Detection of the operating status of the mobile device used.	EXAMINE	The auditor uses practical tests to check the effectiveness of the measures for detecting whether the operating system is outside an operating state that meets the requirements of the operating system manufacturer (e.g. root/jailbreak detection). It also checks whether the application reacts appropriately to the detection. This can be, for example, termination of the application (see O.Resi_5). According to A.OperatingSystem it is assumed that the operating system provides functions with which an application can query the conformity of the operating state with regard to the operating system manufacturer's requirements for the operating system.
O.Resi_3	Detection and prevention of startup in a development/debug environment.	CHECK	The auditor checks the effectiveness of the debug detection through practical tests. (compare O.Resi_5).
O.Resi_4	Abort the start of the application in the event of unusual user rights.	CHECK	The auditor checks the effectiveness of the detection through practical tests (cf. O.Resi_5).
O.Resi_5	Integrity check of the mobile device before processing sensitive data.	EXAMINE	The auditor examines which integrity check is performed by the application. If the check is performed by external tools for which the auditor has no source code, it performs a penetration test (see O.Resi_2 to O.Resi_4). An integrity check must cover at least the following aspects: • Use of "custom firmware". • Up-to-dateness of the operating system version. • Presence of suspicious tools or applications on the device.
O.Resi_6	Verification of the authenticity of the backend system before access.	EXAMINE	The auditor checks the effectiveness of the authenticity check (e.g. certificate pinning) by means of practical tests.

Objective	Short version of the objective	Audit depth	Comments
O.Resi_7	Integration of hardening measures before processing sensitive data.	EXAMINE	The auditor checks whether the application performs an integrity check every time the program is started or for sensitive operations. If not, the resulting residual risks are taken into account in the risk assessment.
O.Resi_8	Implement measures against reverse engineering.	EXAMINE	The auditor checks whether strong measures are taken against reverse engineering. “Strong measures” must conceal all strings, file names and internal names of classes and methods within the application that could provide an attacker with information about the program flow. The auditor checks the effectiveness of the protection against reverse engineering through practical tests and documents the process. The auditor assesses the existing residual risks of the implementation with regard to, among other things T.MemoryStructures, T.InfoDisclosure and T.SensitiveData.
O.Resi_9	Consideration of platforms and versions for access control mechanisms.	EXAMINE	The auditor confirms that the implementation of the access measures does not rely solely on the operating system and may therefore be vulnerable to a downgrade attack on the operating system.
O.Resi_10	Robustness against interference.	EXAMINE	The auditor uses source code analysis and practical tests to check whether faults (e.g. in the power supply, internet connection) or operating errors can lead to a loss of data.

5 Security levels and risk analysis

The basis for the audit judgement shall be a documented risk management procedure. As a general reference, BSI-Standard 200-3 [BSI200-3]ISO 27005 [ISO27005] and Annex B of the Common Criteria Evaluation Methodology [CEM] are mentioned. The auditor may use a comparable risk management procedure that is orientated towards an IT security application after consultation.

The auditors carry out a methodical risk analysis, which must include at least the following steps:

1. Completely analyse the security problem - The starting point for the risk analysis is the threats, assumptions and policies of the application (Chapter 2.4). The auditor establishes a complete list of sensitive data that is collected, generated or used in the application. Sensitive data that is processed solely in the backend system is protected by the assumption A.Backend.
2. Determine protection requirements - IT security generally considers the protection requirements in terms of confidentiality, integrity, authenticity and availability. The auditor classifies the respective protection requirements of all processed data, see Appendix B in [ISO27005]. The data within the scope of the Technical Guideline is differentiated according to its criticality (cf. Table 15).
3. Assess risk scenarios - The auditor carries out an assessment of risk scenarios, taking the established countermeasures into account. This requires a documented, holistic approach to the sensitive data of the application, for example [ISO27005] Section 8.3 and Annexes C/D/E.

The evaluation by the auditor takes into account which protective measures are implemented in the product and their effectiveness (for example, measures against brute force attacks on login credentials). Specifications for secure use are also taken into account, provided these are sufficiently explained to the user. The auditor assess whether the security problem is dealt with appropriately based on the difficulty of the attack paths identified. (The difficulty of an attack replaces the probability of occurrence of a risk referenced in ISO 27005).

The following assessment principles are frequently used to assess attacks on mobile applications:

1. Time-based approach - The auditor estimates the time required for an attacker to override the existing countermeasures. The developer assures that a new product version with new key material will be provided before this time expires (e.g. monthly at the latest) and the application is designed in such a way that attacks can only be carried out on the latest product version. In this scenario, exploitation of the attack path is prevented by a timely update.
2. Reactive approach - Here, the auditor analyses the effective combating of risk scenarios by means of proactive monitoring/reaction. For example, operating parameters are recorded and access to sensitive data is prevented if these indicate intentional modifications. Protection mechanisms implemented externally by the manufacturer must also be considered as part of the TR test.

Based on the residual risks identified, the auditor must make a judgement as to the extent to which the safety problem addressed by the Technical Guideline is adequately fulfilled. Table 15 shows the requirements per date. Certification can only be granted if the audit shows that the requirements for all data are met.

This Technical Guideline primarily serves to evaluate applications as described in chapter 1.3.1. In such applications, the damage caused by the loss of health data is often impossible to quantify, partly because once disclosure has taken place, it can no longer be undone. However, applications that are evaluated in accordance with this Technical Guideline may also contain other sensitive data that must be protected against disclosure. The security level of this data may be lower than that of health data (cf. Table 15). The classification of the security levels for the individual data must be agreed with the BSI on a case-by-case basis. Risk assessments based on established standards can be used for this purpose.

Table 15: Requirement based on data criticality

Criticality	Description	Requirement
Very high	A security breach leads to unquantifiable or potentially serious damage for the data owner.	The measures realized are considered effective in eliminating all risk scenarios without residual risks.
High	A security breach leads to high or medium damage for the data owner.	The measures realized significantly reduce the risk scenarios. The auditor must assess the implementation of remaining attacks and document their effects. In individual cases, the residual risk must be presented and may result in restrictions on the use of the certification.
Normal	At most, minor damage may occur.	The measures realized reduce the risk scenarios. The auditor must evaluate the implementation of remaining attacks and disclose residual risks.

Annex A: Protection needs of sensitive data elements

Depending on the application and the criticality of the data processed, different protection requirements may be necessary. Personal data is subject to data protection and may only be processed for a specific purpose and with consent, see section 3.1.1. The sensitivity of processed data elements is determined in the following table.

Table 16: Protection needs of sensitive data elements

Information	Sensitive	Transfer to backend system allowed	Storage outside a secure environment allowed	Remarks
Application data	Yes	Yes	Yes	—
Input data (from external, third-party software, via keyboard or from device sensors)	No, if not specifically included in another category	Yes	Yes	Pre-treatment, including size checks, escape syntax (depending on further processing)
Access data	Yes	Yes	Yes	e.g. salted hashing. Third-party software for authentication is permitted.
Cryptographic keys of the application	Yes	No	No[9]	Use in third party software for cryptography and session handling is permitted.
Aggregated application data e.g. therapeutic reports as PDF	Yes	Yes	Yes	A display may only take place with an integrated viewer. The implementation should avoid storage on the device. Storage is only permitted in encrypted form. The discharge required for the form of therapy should take place via a secure channel.

[9] Except for public keys or cryptographic keys of third-party software, provided they are not under the control of the application developer and mobile terminals that do not have a secure environment (e.g. embedded Secure Element/Secure Enclave/Trusted Execution Environment) are excluded.

Information	Sensitive	Transfer to backend system allowed	Storage outside a secure environment allowed	Remarks
Biometric data	Yes	No	No[10]	Biometric data is not accessible to the application, except for a reference feature (e.g. User 1).
Public certificates and information for certificate pinning	No	Yes	Yes	—

[10] Automatically fulfils when the operating system does not provide these data to the application, otherwise the data as sensitive data must be protected by the application

List of abbreviations

API	Application Programming Interface
App	application
A.*	Assumption
BSI	Federal Office for Information Security
GPS	Global Positioning System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and communication technologies
iOS	Apple’s operating system for mobile devices
IoT	Internet of Things
IPC	Inter-process communication
O.*	Objective (audit aspect)
OSP.*	Organizational Security Policies
R.*	Recommendation
SDK	Software Development Kit
SGB V	Social Security Code (SGB) Fifth Book (V)
SGB XI	Social Security Code (SGB) 11th Book (XI)
SMS	Short Message Service
SPD	Security problem definition
SSID	Service Set Identifier
T.*	Threat
TR	Technical Guideline
TLS	Transport Layer Security

URL	Uniform Resource Locator
WiFi	Wireless Fidelity
WIFI	Wireless Local Area Network
XML	Extensible Markup Language

References

[AIS20]

Federal Office for Information Security, “AIS 20: Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren”, 15/05/2013, available at https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_pdf.pdf?__blob=publicationFile&v=1

[BMG-EH]

Federal Ministry of Health, “Glossar: E-Health, Version 2023”, available at https://www.bundesgesundheitsministerium.de/service/begriffe-von-a-z/e/e-health.html

[BSI200-3]

Federal Office for Information Security, “BSI-Standard 200-3 Risikoanalyse auf der Basis von IT-Grundschutz”, Version 1.0, available at https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/BSI_Standards/standard_200_3.pdf?__blob=publicationFile&v=2

[CEM]

“Common Methodology for Information Technology Security Evaluation – Evaluation methodology”, April 2017, Version 3.1, Revision 5, available at https://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R5.pdf

[DIGAV]

“Verordnung über das Verfahren und die Anforderungen zur Prüfung der Erstattungsfähigkeit digitaler Gesundheitsanwendungen in der gesetzlichen Krankenversicherung (Digitale Gesundheitsanwendungen-Verordnung - DiGAV)”, Version 2023, available at https://www.gesetze-im-internet.de/digav/__4.html

[DFS]

Google Developers, “Design for Safety”, Version 2023, available at https://developer.android.com/quality/security-and-privacy

[GDR18]

we are social, “Global Digital Report 2018”, Version January 2018, available at https://wearesocial.com/de/blog/2018/01/global-digital-report-2018

[gemSpec_IDP_Sek]

gematik, “ Spezifikation Sektoraler Identity Provider”, available at https://gemspec.gematik.de/docs/gemSpec/gemSpec_IDP_Sek/latest/

[iOSSF]

Apple Inc. “iOS Security Framework”, available at https://developer.apple.com/documentation/security

[ISO27005]

BS ISO/IEC 27005:2011, Information technology – Security techniques – Information security risk management

[KCC-C5]

Federal Office for Information Security, “Kriterienkatalog Cloud Computing”, Version 2020, available at https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/CloudComputing/Anforderungskatalog/2020/C5_2020.pdf?__blob=publicationFile&v=2

[MASVS]

The OWASP Foundation, “Mobile AppSec Verification Standard”, Version 2.0, available at https://github.com/OWASP/owasp-masvs/releases/tag/v2.0.0

[MSTG]

The OWASP Foundation, “Mobile Security Testing Guide”, Version 1.6, available at https://github.com/OWASP/owasp-mastg/releases/tag/v1.6.0

[NIST80057]

National Institute of Standards and Technology, “Recommendation for Key Management”, Revision 5, available at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf

[SGBV33a]

Bundesanzeiger, “Social Code (SGB) Fifth Book (V) –Gesetzliche Krankenversicherung – § 33a Digitale Gesundheitsanwendungen” Version 2023, available at https://www.gesetze-im-internet.de/sgb_5/__33a.html

[SGBXI40a]

Bundesanzeiger, “Social Code (SGB) 11st Book (XI) – Soziale Pflegeversicherung – § 40a Digitale Pflegeanwendungen”, Version 2023, available at https://www.gesetze-im-internet.de/sgb_11/__40a.html

[SSDG]

European Union Agency For Network And Information Security, “Smartphone Secure Development Guidelines”, Version December 2016, available at https://www.enisa.europa.eu/publications/smartphone-secure-development-guidelines-2016/at_download/fullReport

[TR02102-1]

Federal Office for Information Security, “Kryptographische Verfahren: Empfehlungen und Schlüssellängen”, Version 2023-01, available at https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile&v=8

[TR02102-2]

Federal Office for Information Security, “Kryptographische Verfahren: Empfehlungen und Schlüssellängen Teil 2 – Verwendung von Transport Layer Security (TLS)”, Version 2023-01, available at https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2.pdf?__blob=publicationFile&v=6

[TR03107-1]

Federal Office for Information Security, “ Elektronische Identitäten und Vertrauensdienste im E-Government Teil 1”, Version 1.1.1, available at https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03107/TR-03107-1.html

[TR03116-4]

Federal Office for Information Security, “ Kryptographische Vorgaben für Projekte der Bundesregierung Teil 4: Kommunikationsverfahren in Anwendungen”, 7 March 2023, available at https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03116/BSI-TR-03116-4.pdf?__blob=publicationFile&v=5

[TR03161-2]

Federal Office for Information Security, “Requirements for Healthcare Applications Part 2: Web applications”, Version 2.0, available at https://www.bsi.bund.de/dok/TR-03161-2

[TR03161-3]

Federal Office for Information Security, “Requirements for Healthcare Applications Part 3: Backend systems”, Version 2.0, available at https://www.bsi.bund.de/dok/TR-03161-3