Federal Office
for Information Security

Technical Guideline TR-03161: Requirements for Healthcare Applications

Part 2: Web Applications

Version 2.0

Change history

Version	*Date*	*Name*	*Description*
2.0	11.09.2024	Unit D 24	Translation of German version 2.0

Federal Office for Information Security
PO Box 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.

This TR is aimed at developers of web applications in the healthcare sector. In addition, it can be considered as a guideline for web 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 to "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, web applications store sensitive and personal data, from pulse frequency, sleep rhythm records, medication schedules and medical prescriptions. Web applications connect the user with multiple services and therefore act as communication hubs. A compromised application 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 web applications, can decrease the risk and may even prevent it at all. Already during the development phase, manufacturers should plan very responsibly how a web application 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 web application developers in the creation and operation of secure software solutions for healthcare applications. If the application relies on

functionalities of a backend system, the security of the backend system is also essential for a full safety assessment (see Chapter. 2.3).

1.3 Overview of the Technical Guideline

1.3.1 Methodology

Applications within the scope of this TR are web applications in the healthcare sector. This includes in particular digital health applications within the meaning of § 33a SGB V (see [SGBV33a]) and 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 platforms, this Technical Guideline makes no requirement for completeness. Instead it can be considered as a minimum requirement for the secure operation of a web application.

The Technical Guideline formulates a security problem definition (SPD) which has potential threat scenarios. From this SPD, testing aspects for web applications and their platforms or operational environments are derived from the SPD in order to protect against these threats.

The threat scenarios and test aspects formulated in this Technical Guideline are based on experience gained by the BSI in previous inspections of web applications in the healthcare sector. It follows international standards, such as the Application Security Verification Standard [ASVS], and the Web Security Testing Guide [WSTG].

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 Guideline uses the following terms:

Table 1: Terms of the Technical Guideline

Term	Description
MUST	The application must necessarily have a certain property.
MUST NOT/DO NOT NEED(S)	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 secure operation or this property 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 § 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 devices

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 mobile applications go far beyond the use of web applications on mobile browsers or hybrid applications, this publication focuses solely on web applications and the web part of hybrid approaches. For additional guidance on the secure development and operation of native applications, the BSI recommends “TR-03161 Requirements for Healthcare Applications Part 1: Mobile applications” [TR03161-1].

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 application benefit from the closeness to the operating system, therefore they 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 adjustment 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 operations. 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 and web applications. 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 an 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 (webView) 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.

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 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 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 web applications, but reflects the state of the art for almost all applications. Three main scenarios for using backend systems are distinguish:

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

Depending on the type of operation and the associated different attack vectors, the application developer has different possibilities to ensure the security of the overall application and of the data stored and processed.

It is not easy to separate the web application from the backend system, as the web application is delivered by the backend system for use by the user on their platform with a web browser. However, this also makes it clear that a web application is inconceivable without a backend system. This means that in addition to this Technical Guideline, "TR-03161 Requirements for Healthcare Applications Part 3: Background systems" [TR03161-3] must always be taken into account in such an architecture.

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 operating 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 customers, 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 manufacturers 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 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 providers 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.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.Browser	The web browser used by the user is free of vulnerabilities. It is therefore assumed that access to unprotected data structures in the memory, which enable access to keys and sensitive data, for example, is excluded. The web browser implements the TLS protocol to secure communication with the backend system in accordance with the current state of the art in a secure manner. This includes a correct and complete certificate check.
A.Device	The device running the application is operated by the user itself and protected from vulnerabilities.

A.DevRNG	It is assumed that the user's web browser uses randomness in a sufficient quality to establish a secure connection to the background system, so that the confidentiality and integrity of the transmitted data is ensured.
A.Updates	The platform consisting of the operating system and web browser on which the web application is used is operated by the user and protected against vulnerabilities, for example by updating the operating system and web browser after updates have been made available. Its security has not been deliberately compromised by the user (e.g. “roots” or “jailbreaks”[2]). The web browser used is obtained exclusively from official and trustworthy sources.
A.User	The user of the web application checks in the browser whether they are connected to the correct Internet address. The user uses the standard security settings of the web browser used. *Application note*: In the case of the web application, there is a priori no way to display information to the user authentically. The web application relies on the user using the correct website via the web browser. A fake website can also offer a valid TLS connection and will then simply withhold or falsify the necessary security information from the user. Therefore, the establishment of the initial trust anchor via server authentication as part of the TLS connection, especially with the correct web presence, can only be formulated as an assumption. If there is an independent channel between the provider of the web application and the user for transmitting information, e.g. via a health insurance company, this can of course be used to provide the user with authentic information about the provider's web presence.

2.4.2 Threats

T.SensitiveData	Sensitive data in the Technical Guideline are to be understood as defined in Annex A. An unauthorized person gains access to such sensitive data in the web application, such as unencrypted data stored in the file system (e.g. browser cache) or RAM. This also means that an attacker can access encrypted, sensitive data in plain text after analyzing the encryption mechanism between the web browser and the backend system.
T.Auth	An attacker gains access to sensitive data of other users using a different user ID or a different role or group membership.
T.DevFunctions	An attacker uses hidden or remaining developers or debugging functions in the web application to subvert security measures.
T.Expense	The web application causes unforeseen, additional costs for the user or operator.
T.Impersonation	An attacker is gains unauthorized access to sensitive data or chargeable functions of another user by missing or faulty access controls or by guessing access parameters.
T.InfoDisclosure	An attacker analyzes the web 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 the memory or via the transport route without being noticed.
T.VisibleAsset	The attacker can read sensitive data displayed on the web application by “shoulder surfing”[3].

[2] The weakening of the operating system own security functionalities by obtaining increased access rights and enabling the installation of applications from unknown sources.

[3] For 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.BrowserCred Current platforms with web browsers generally offer the option of storing login information for websites securely in order to be able to offer users simplified access when they visit the website again. The web application appropriately informs the user of the associated residual risks.

OSP.User The user must be made aware of his obligations to co-operate via the terms of use of the web application. If the user does not agree to these terms of use, he MUST be excluded from using the service. If, on the other hand, the user agrees, it is assumed that the user will behave in accordance with the terms of use.

OSP.CriticalUpdates The manufacturer shall permanently check and monitor the browsers that can be used for the web application and the third-party software[4] used for the web application for exploitable vulnerabilities. If vulnerabilities are published or discovered, the manufacturer must provide an update for the web application at short notice, which makes the vulnerability inaccessible or reduces its exploitability. The backend system MUST prevent the application from being used with an outdated browser or browser in an outdated version.

OSP.DataSovereignty The web application ensures the user's data sovereignty. The web application informs the user of existing risks due to the configuration of their device and allows them to decide to cancel use. At the user's request, the web application deletes data already recorded in the background system and informs the user that any locally stored data (e.g. stored form data or browser cache) must be deleted by the user. If the user exports sensitive data in unencrypted form and thus prevents it from being monitored by the web application, the web application will inform the user that the user is responsible for the data security of this exported 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.Purpose Any data collection, processing, storage and forwarding may only take place with a purpose limitation. The developer shall publish the lawful purpose of the web 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.

Method of administration: Location data such as IP addresses, GPS, etc. may only be processed if they are essential for the functioning of the web application. The data thus collected shall be processed solely for purpose. They shall not be persistent directly or indirectly in the device unless directly necessary for the intended use. *Application note*: Location data such as IP addresses, GPS etc. may only be processed if it is essential for the function of the web application. The data collected in this way may only be processed for

[4] A third-party software is intended to understand the summary of functionalities that have not emerged in the sovereignty of the application developer and which is also not part of the functionality of the operating system platform used.

a specific purpose. They may not be persisted directly or indirectly in the device unless this is directly required by the processing purpose.

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

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. A breach of confidentiality may also be possible without overcoming separation mechanisms if a malicious client on the same physical machine manages to exploit side channels that may arise when processing sensitive data in another virtual machine.

Communication connections between the web application and the background system are protected using the cryptographically secured TLS protocol. In this scenario, the TR assumes one-sided authentication, whereby the web browser used checks the authenticity of the background system. 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 the connection setup is that the attacker can fake the authenticity of their own messages. As a result, the attacker could view and manipulate sensitive data transmitted from the application to the background system. Unlike in the case of the native application, there is no possibility for a web application to introduce additional randomness into the connection in order to mitigate this residual risk.

With the assumption A.Browser , it was assumed in the sense of a closed security statement in this Technical Guideline that the web browser used is free of vulnerabilities. In reality, this cannot be completely assumed. Some test criteria attempt to minimize this residual risk, in particular e.g. O.Arch_8. Nevertheless, the residual risk of vulnerabilities in the web browser used cannot be completely ruled out.

Other residual risks are that the user utilizes certain features of the platform or web browser, which in turn can be associated with a risk to sensitive data. One example of this is O.Auth_9. These or similar verification aspects should ensure that the user is fully informed of their responsibilities via the terms of use.

In general, due to the limitations described in Chapter 2.1 and 2.2 in Part 1 of this family of Technical Guidelines [TR03161-1], it is not possible to make an overall statement about the security of the mobile web application, even taking into account all the test aspects listed. In order to increase the security of the entire web 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 healthcare 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 healthcare applications. 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 web application and the processing of personal data (e.g. in the terms and condition of use) and inform the user of this at least when the application is first used.
O.Purp_2	The web application MUST NOT collect and process data that does not serve the legitimate purpose of the application.
O.Purp_3	The web 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 web application or the backend system.
O.Purp_5	The web 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 web application uses third-party software, all functions used MUST be necessary for the legitimate purposes of the web application. The web 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 a web application, sensitive data MUST NOT be shared with third parties. The web application MUST fully inform the user of the consequences of any sharing of web application data that serves the primary or legitimate purpose and obtain the user's consent (OPT-IN).

O.Purp_9 The web 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 web application and backend system.

O.Arch_2 Already in the design phase of the web application and the background system, it MUST be taken into account that the application will process sensitive data in the production phase. The architecture of the application MUSST ensures the secure collection, processing, storage and deletion of the sensitive data in a data life cycle.

O.Arch_3 The life cycle 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].

O.Arch_4 Sensitive data stored in backups MUST be encrypted according to the current state of the art. This includes the persistence of sensitive data by the browser, for example in its cache.

O.Arch_5 If the web application uses third-party software, the developer MUST ensure[5] 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[5] that unused functions cannot be activated by third parties.

O.Arch_6 The architecture of the web application SHOULD follow a minimalist approach and be realized with a server-side localized processing logic, i.e. no complex active content (Java applets, ActiveX plugin, etc.) SHOULD be used.

O.Arch_7 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_8 The web application MUST check that the web browser used is up-to-date when it is started. If a security-relevant update has not yet been installed, the web application MUST NOT allow access to sensitive data.

O.Arch_9 The web application SHOULD use HTTP server headers that correspond to the current state of the art and increase the security of the application. These include HTTP Strict Transport Security (HSTS), Content Security Policy (CSP) and X Frame Options.

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.

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

O.Source_3	Error messages and log files MUST NOT contain sensitive data (e.g. user identifiers or session IDs).
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 web application SHOULD cancel access to sensitive data and securely delete it from memory.
O.Source_6	All options to support development (e.g. developer URLs, test methods, remnants of debug mechanisms, etc.) MUST be completely removed in production.
O.Source_7	Modern security mechanisms such as obfuscation and stack protection SHOULD be activated to build the application.
O.Source_8	Tools for static code analysis SHOULD be used for the development of the application.
O.Source_9	If the web application uses URL redirects, this MUST be done in a controlled manner.
O.Source_10	The web application MUST provide mechanisms to prevent functionalities that are not within the manufacturer's development sovereignty from being injected into the web application and executed.
O.Source_11	Sensitive data MUST NOT be included in the URL. The web application MUST process such data in HTTP request headers or POST parameters.

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_6) 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 web application or the backend system based on the criticality of exploitable vulnerabilities. After the grace period has expired, the web application MUST NOT be offered for use.
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.

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

3.1.5 Objective (5): Cryptographic implementation

O.Cryp_1	When using encryption in the web 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]).

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 IP address 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).

O.Auth_16 If the login credentials are changed, the user SHOULD be informed of the change via the last valid contact details stored. In this way, the user SHOULD be offered the option of blocking the reported change and setting new login credentials after authentication.

O.Auth_17 The user MUST be made aware of the residual risk associated with the storage of login credentials in the web browser or another external application for a more comfortable login process in the terms of use of the web application.

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 The changing and resetting of passwords MUST be logged.

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 web application MUST provide maximum security.

O.Data_2 If the user exports sensitive data without encryption, the web application MUST inform the user that the user is responsible for the data security of this exported data.

O.Data_3 The web application MUST NOT make resources that allow access to sensitive data available to third parties.

O.Data_4 All sensitive data collected MUST NOT be kept in the application beyond the duration of their respective processing.

O.Data_5 The application MUST comply with the principles of data minimization and purpose limitation.

O.Data_6 Sensitive data SHOULD be stored and processed in the backend system.

O.Data_7 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_8 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_9 When entering sensitive data via the keyboard, the application SHOULD prevent recordings from becoming recognizable to third parties.

O.Data_10	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_11	Sensitive data such as biometric data or private keys MUST NOT be exported from the component on which they were generated.
O.Data_12	The web application cannot prevent third parties from accessing and saving the screen (e.g. screenshots and views for app switching). The user MUST be informed via the terms of use that sensitive data can be compromised via screenshots or views for app switching.
O.Data_13	The terms of use of the web application MUST inform the user of the risk associated with the fact that the connection to the background system is still open when the end device is locked if the user has not explicitly logged out.
O.Data_14	The web application SHOULD ensure that all sensitive data and application-specific login information is no longer accessible in the web browser when it is terminated. This includes cookies and web storage in particular.
O.Data_15	The web application MUST give the user the option to have all sensitive data and application-specific login information completely deleted or made inaccessible upon final termination.
O.Data_16	The HTTP-only flag MUST be used for all cookies that are not accessed via JavaScript.
O.Data_17	The secure flag MUST be set for all cookies that contain sensitive data.
O.Data_18	The autocomplete function MUST be switched off for all form fields with sensitive input data.
O.Data_19	Data persisted in the browser SHOULD be unreadable for other hosts of a domain (i.e. avoidance of domain cookies).

3.1.8 Objective (8): Paid resources

O.Paid_1	The web 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 web application MUST obtain the user's consent before requesting access to paid resources.
O.Paid_4	The web application MAY obtain the user's permanent consent for access to frequently used paid resources or paid services.
O.Paid_5	The web 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 web 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 web 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 MUST 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 web application MUST be encrypted end-to-end with mutual authentication.

O.Ntwk_2 The configuration of the TLS connections MUST comply with the current state of the art (see [TR02102-2]).

O.Ntwk_3 The web application MUST use the security functionality of the operating system and browser used in order to establish secure communication channels.

O.Ntwk_4 The web application SHOULD prevent the use of certificates whose certificate chain does not appear trustworthy to the manufacturer.

3.1.10 Objective (10): Platform-specific interactions

O.Plat_1 To use the web 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 web application MUST NOT request permissions that are not necessary for the fulfillment of its primary purpose.

O.Plat_3 The web application MUST NOT request permissions that are not necessary for the fulfillment of its primary purpose.

O.Plat_4 The web 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 web application MAY offer the option of displaying messages and notifications the user, including those containing sensitive content. This MUST be deactivated by default.

O.Plat_6 The web application MUST restrict the reloading of content to sources that are under the manufacturer's control or have been authorized by the manufacturer.

O.Plat_7 The web application MUST inform the user of the risk that user-specific data may remain in the RAM after the web application is closed.

O.Plat_8 The user MUST be informed about security measures, as long as they can be realized by the user.

3.1.11 Objective (11): Resilience

O.Resi_1 The web application MUST provide the user with accessible best practice recommendations for the safe use of the application and its configuration.

O.Resi_2 The web application MUST inform the user via the terms of use of the risks that exist for the user's data when using devices whose operating system is not in an operating state intended by the operating system manufacturer.

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 evaluator 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 evaluator 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 evaluator requires these for a comprehensive assessment.
EXAMINE	The evaluator examines (analogue to the use of terms in Common Criteria Evaluation Methodology) the test characteristic in question. The evaluator 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 evaluator.

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 verifies that all data processed corresponds to the legitimate purposes of the application. 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_2with 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 evaluator 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. The auditor also checks whether a history of this directory can be requested.
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 primary or lawful purposes.	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. The use by third parties could, for example, consist of calling up cookies, banners from third parties or the redirection of form input via websites of uninvolved parties.
O.Purp_9	Only display sensitive data on the screen for a specific purpose.	CHECK	The auditor examines the developer's 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). In the case of forms, the web application must instruct the web browser to hide the input of sensitive data (type='password') and to handle input appropriately.

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.
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 persisted in backups or in the browser cache.	EXAMINE	The auditor uses source code analysis and practical tests to examine whether sensitive data is present in unencrypted in backups and/or in the browser cache.
O.Arch_5	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_6	Localized processing logic of the web application on the server side.	EXAMINE	The auditor checks whether the processing logic of the web application is realized by the background system. It ensures that no Java applets, ActiveX plugins or comparable complex active content is used. When dealing with active content, the “Sicherheitsmaßnahmen beim Einsatz aktiver Inhalte” [BSI-CS-120] must be observed.
O.Arch_7	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.

Objective	Short version of the objective	Audit depth	Comments
O.Arch_8	Check that the web browser used is up-to-date.	CHECK	The auditor checks whether access to sensitive data is possible with an outdated browser version. The background system can use information in the HTTP header to check which version of the web browser is being used when the web browser establishes a connection. However, the backend system relies on the cooperation of the web browser and cannot authentically determine this information. It can be assumed that the user is not using an unauthorized web browser with a falsified browser ID as they are complying with the terms of use (see OSP.User).
O.Arch_9	Use of state-of-the-art HTTP server headers.	CHECK	The auditor checks whether appropriate HTTP server headers are used. If no state-of-the-art HTTP server headers are used, this must be taken into account in the risk assessment.

4.3.3 Audit characteristics for objective (3): Source code

Table 6: Audit characteristic: Source code

Objective	Short version of the objective	Test depth	Comments
O.Source_1	Checking inputs before processing.	CHECK	The auditor checks whether all inputs are checked according to the state of the art before they are processed. 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	Test 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. 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, notifications.
O.Source_4	Controlled treatment and documentation of exceptions.	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	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_7	Use of modern security mechanisms.	EXAMINE	The auditor examines whether modern security mechanisms were used during the development and deployment of the application.

Objective	Short version of the objective	Test depth	Comments
O.Source_8	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.
O.Source_9	Controlled use of URL redirects.	EXAMINE	The auditor ensures that URL redirects are used in a controlled manner. Under no circumstances may the link to an external page contain the session ID. If the list of redirect URLs is known, only these may be redirected to. If possible, an indexed list can also be kept on the server side in these cases and the redirect URL can then be determined via an index. A redirect can also take place via a website explicitly displayed to the user so that the user can check the link themselves before actively clicking on it. Local redirects must be checked to ensure that the target URL does not lead to an external page. If avoidable, it should not be possible for the user to enter the redirect URL themselves. If it is unavoidable that a user may enter the redirect URL themselves, it must be comprehensively checked to ensure that it is valid, appropriate for the web application and permitted for the user. If implemented, whitelisting of authorized addresses should be preferred over blacklisting.

Objective	Short version of the objective	Test depth	Comments
O.Source_10	Preventive measures to protect functionalities outside the developer's own sovereignty.	EXAMINE	The auditor uses source code analysis and practical tests to ensure that the web application takes state-of-the-art measures to prevent the execution of injected functionalities. For example, as a countermeasure against cross-site request forgery (CSRF) attacks, the web application can insert a session code (also known as a synchronizer token pattern (STP) or CSRF token) into each URL as an additional random (i.e. unguessable) parameter in a hidden field. Based on the session code, the backend system can also check the validity of the request.
O.Source_11	No sensitive data in the URL.	EXAMINE	The auditor uses source code analyses and practical tests to ensure that sensitive data is only processed using secure, state-of-the-art methods and is not included in the URL at any time.

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

Table 7: Audit characteristic: Third-party software

Objective	Short version of the objective	Test 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.
O.TrdP_2	Use of the 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.

Objective	Short version of the objective	Test depth	Comments
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.
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): Cryptographic

Table 8: Audit characteristic: Cryptographic

Objective	Short version of the objective	Test depth	Comments
O.Cryp_1	No hard-coded keys or other secrets.	EXAMINE	The auditor examines whether hardcoded secret or private keys are 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 checks the strength of the keys used against the current state of the art (cf. [TR02102-1]).

4.3.6 Audit characteristics for objective (6): Authentication

Table 9: Audit characteristic: Authentication and authorization

Objective	Short version of the objective	Test depth	Comments
O.Auth_1	Developer concept for the authentication 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 methods used on the basis of the current state of the art.

Objective	Short version of the objective	Test depth	Comments
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 or the measures taken are not exclusively enforced by the backend system, the developer's considerations must be checked and 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.
O.Auth_4	Authentication via additional procedures corresponding to a lower level of security.	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.

Objective	Short version of the objective	Test depth	Comments
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.
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).

Objective	Short version of the objective	Test depth	Comments
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 evaluator checks whether the application supports authentication of the backend system.
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.

Objective	Short version of the objective	Test depth	Comments
O.Auth_16	Notification of the user when access parameters are changed.	CHECK	The auditor checks whether the user is informed via the most recently stored, valid contact data when the access parameters are changed. It also ensures that the user is given the opportunity to block the change and set new access parameters via this mechanism. If the access parameters are changed by an attacker, the user who is actually authorized can use this mechanism to prevent any major damage by blocking the new access parameters and thus preventing the attacker from gaining access. Special attention must therefore be paid to the necessary re-authentication of the authorized user.
O.Auth_17	Informing the user about the residual risk of storing login credentials in the web browser.	CHECK	The auditor checks whether the user is informed in a simple and understandable form about the residual risk of storing login credentials in the web browser or another external 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 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 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	Test depth	Comments
O.Pass_5	Use of cryptographically safe 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

Audit aspect	Short version of the audit aspect	Test 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.
O.Data_2	Encryption of all sensitive data in exports.	CHECK	The evaluator checks whether the user can export sensitive data from the application unencrypted. If this is the case, the evaluator verifies whether the user is appropriately made aware of the resulting risks.
O.Data_3	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.

Audit aspect	Short version of the audit aspect	Test depth	Comments
O.Data_4	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.
O.Data_5	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_6	Storage and processing of sensitive data.	CHECK	The auditor checks which data the web 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_7	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 background system.
O.Data_8	Access restriction when collecting sensitive data.	EXAMINE	The auditor examines whether collected sensitive data is made available to other applications on the device or whether data is stored in public directories.
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. 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.

Audit aspect	Short version of the audit aspect	Test depth	Comments
O.Data_10	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.
O.Data_11	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 private cryptographic keys.
O.Data_12	Informing the user about the risks of saving screenshots.	CHECK	The evaluator checks whether the user is adequately informed about the risks of screen capture.
O.Data_13	Informing the user about the risks of an active connection to the backend system when the end device is locked.	CHECK	The auditor checks whether the user is informed properly about the risks of an active connection to the backend system when the end device is blocked.
O.Data_14	Remove or make all sensitive data inaccessible in the browser when the web application ends.	CHECK	The auditor checks whether data remains in the browser after the web application has ended. 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.
O.Data_15	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_16	HTTP-only flag for cookies.	CHECK	The auditor ensures that the HTTP-only flag is set for all cookies that are not accessed using JavaScript.
O.Data_17	Secure flag for cookies.	CHECK	The auditor ensures that the secure flag is set for all cookies that contain sensitive data.

Audit aspect	Short version of the audit aspect	Test depth	Comments
O.Data_18	Autocomplete function for form fields.	CHECK	The auditor ensures that the auto-complete function is deactivated for all form fields in which sensitive data is entered.
O.Data_19	Avoidance of domain cookies.	CHECK	The auditor ensures that persisted data is not readable for other hosts of a domain.

4.3.8 Audit characteristics for objective (8): Priced resources

Table 11: Audit characteristic: Priced resources

Objective	Short version of the audit aspect	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 priced resources or paid services.	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 audit aspect	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 paid 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 audit aspect	Test 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 state of the art.	EXAMINE	The auditor validates that the data specified in O.Ntwk_1 corresponds to (see [TR02102-2]) the state of the art. This also includes methods that prevent any transmission of data outside the TLS channel. For example, HTTP Strict Transport Security (HSTS) can be used to ensure that only HTTPS is used between the browser and the background system providing the web application. Content Security Policy (CSP) can also be used to ensure that content can only be loaded from a defined list of sources.
O.Ntwk_3	Secure communication channels only with operating system functions.	EXAMINE	The auditor examines whether operating system functions are used to establish secure communication channels. Alternatively, third-party software that fulfils the requirements described in section 3.1.4 can be used. Proprietary implementations for establishing up secure communication channels are not permitted.
O.Ntwk_4	Verification of certificate chains.	EXAMINE	The auditor examines the functionality of the certificate check used through practical tests and source code analyses. If the manufacturer has not implemented a certificate check, the evaluator examines the manufacturer's considerations regarding the effects of a certificate validation on the confidentiality of the data and the availability of the application. The resulting residual risks must be documented in the risk analysis.

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.
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).
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.
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	Reload active content.	CHECK	The auditor checks whether the components prevent the reloading of active content or restrict it to sources under the developer's control. The selection of authorized sources is taken into account in the risk analysis.

Objective	Short version of the objective	Audit depth	Comments
O.Plat_7	Information of the user about the storage of sensitive data in the RAM.	CHECK	The auditor checks whether the user is adequately informed about the possibility of sensitive data being stored in RAM and about the resulting risks. This duty to inform includes the option from O.Plat_8.
O.Plat_8	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.

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.
O.Resi_2	Information of the user when using devices with limited security features.	CHECK	The auditor checks whether the user is informed appropriately about the risks of using devices whose operating system is not in an operating state intended by the operating system manufacturer.

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 web 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. Table15). 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 safe 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[7]	Use in third party software for cryptography and session handling is permitted.
Aggregated application data e.g. therapeutic report 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.
Public certificates	No	Yes	Yes	—

[7] Exceptions are public keys or cryptographic keys from third-party software if these are not under the control of the application developer and mobile devices that do not have a secure environment (e.g. embedded Secure Element/Secure Enclave/Trusted Execution Environment).

List of abbreviations

Table 17: List of abbreviations

App	Route of administration
A.*	Assumption
BSI	Federal Office for Information Security
CSP	Content Security Policy
CSRF	Cross-Site-Request-Forgery
EU	European Union
GPS	Global Positioning System
HSTS	HTTP Strict Transport Security
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
IP	Internet Protocol
IT	Information technology
O.*	Objective (audit aspect)
OSP.*	Organisational Security Policies
PDF	Portable Document Format
SDK	Software Development Kit
SGB V	Social Security Code (SGB) Fifth Book (V)
SGB XI	Social Security Code (SGB) 11 Book (XI)
SMS	Short Message Service

SPD	Security problem definition
T.*	Threat
TR	Technical Guideline
TLS	Transport Layer Security
URL	Uniform Resource Locator
WIFI	Wireless Local Area Network

References

[ASVS]

The OWASP Foundation, “Application Security Verification Standard”, Version 4.0, available at https://www.owasp.org/images/d/d4/OWASP_Application_Security_Verification_Standard_4.0-en.pdf

[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

[BMG-EHI]

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

[BSI200-3]

Federal Office for Information Security, “BSI-Standard 200-3 Risk analysis based on 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[BSI-CS-120]

Federal Office for Information Security, “Sicherheitsmaßnahmen beim Einsatz aktiver Inhalte - Verwendung aktiver Inhalte durch Anbieter von Webanwendungen”, Version 3.0, available at https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_120.html

[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

[GDR18]

we are social, “Global Digital Report 2018”, Version Januar 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/

[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

[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

[RFC7469]

C. Evans, C. Palmer, R. Sleevi, Google Inc., “Public Key Pinning Extension for HTTP”, Version April 2015, available at https://tools.ietf.org/html/rfc7469

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

[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

[TR02102-1]

Federal Office for Security in 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=10

[TR02102-2]

Federal Office for Security in 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=2

[TR03161-1]

Federal Office for Information Security, “Requirements for Healthcare Applications Part 1: Mobile applications”, Version 3.0, available at https://www.bsi.bund.de/dok/TR-03161-1

[TR03161-3]

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

[WSTG]

The OWASP Foundation, “Web Security Testing Guide”, Version 2021, available at https://owasp.org/www-project-web-security-testing-guide/stable/