SUPPORT POOL OF EXPERTS PROGRAMME

Standardised Messenger Audit
D2 - Audit Methodology

by Mathieu CUNCHE

Professor at INSA-Lyon, CITI-Lab
Faculty member of the Inria Privatics team

To produce this deliverable the expert worked closely with the BfDI (ie the DE federal SA), which provided significant input and feedback.

Contributors: Thore Hendrikson, Aline Sylla and Nina Zinnhobler

Document submitted in May 2024

Table of Contents

1 Introduction

This Document - D2 - features the audit methodology. For each requirement listed in D1, that is not labelled as being out of scope, an audit methodology is listed in this document. D2 follows D1 in form and sequence.

To make working with the methodology easier, the tests are graded as either ”basic” or ”intermediate”, whereas intermediate level tests require more expertise and possibly tools to carry out than basic tests. Some requirements may feature multiple test cases of different levels, but a basic level test is provided for all requirements. The structure of the test is as follows: The test is always preceded by a superordinate clause featuring the wording of the requirement, e.g. „The messenger MUST feature a privacy policy.“ The respective test criterion then passes through the processes (1) preliminary steps/perquisite, (2) verification steps, and (3) validation. The preliminary or prerequisite check lists necessary conditions and requirements that must be met in order to perform the verification. The verification steps are then used to determine the current state of the application in order to make an informed assessment in the next step. The validation passage is used to determine if the application passes or fails the test, or if the requirement is not applicable to the tested application. Optional subsequent steps may follow.

It is important, that the auditor takes care to evaluate the context and functionalities of the messenger they are auditing and adjust their audit accordingly. This catalog is not exhaustive, but covers common features, as they may be found in some or many messengers. As the messenger services industry is constantly evolving, new features, contexts and configurations may arise, or are already in place, but not covered in this document. For example, a messenger could have a permissions model, where different account types have different permissions regarding different functionalities of the messenger. In this case the auditor could take additional steps to not only verify certain requirements for all available frontends, but also for all available account types.

1.1 Types of verification

Several types of verification are found in this document:

- UI-based: verification is done by interacting with the application through the user interface.
- Information request: verification is done by requesting information to the controller and then analysing this information.
- System & Network analysis: verification is done by passively/actively analysing system and network activity of the application.

1.2 Verification level

- Basic: verification requires minimal skills and simple tools (end-device and Internet connectivity).
- Intermediate: verification requires basic prior knowledge and technical understanding and/or dedicated tools (networking hardware and auditing software).
- (For future versions) Advanced: verification requires advanced prior knowledge and extended technical understanding and/or dedicated tools.

1.3 Standard structure for verification

REQ_X	`level`
Reminder of the requirement	
Prerequisite	
Necessary conditions and requirements that must me met in order to perform the verification.	
Verification steps	
Steps the auditor performs in order to determine the current state of the application under test and/or to acquire all essential information in order to be able to make an informed assessment.	
Validation	
Explicit definition on how to make the assessment if the requirement is not applicable (NOT APPLICABLE), fulfilled (PASS) or failed (FAIL) based on the information acquired during the verification.	
(optional) Subsequent steps	
Possible information and effects (eg. not applicable) on other requirements.	

1.4 General instructions

1.4.1 Verification from other accounts

For some requirements, the verification concerning an account A needs to be performed using the *point of view* of another account. This account can be any account or an account that has a link with A: they are mutual contacts, they exchanged messages, they are part of the same group/community ... To this aim a *populated* account (see section A.1.4) is accompanied with a set of linked accounts.

2 Audit Methodology for Requirements for a a GDPR compliant messenger

2.1 Lawfulness of processing (Art. 6 GDPR)

LEG_BASIS_1 `basic`

The processor of personal data MUST provide a legal basis for the processing.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all personal data processed by the application;
2. Ask the controller to provide the legal basis for the processing of each listed personal data;
3. Verify, for each personal data processed, that the controller has provided a valid legal basis;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller has provided a valid legal basis for all personal data processed. Otherwise, the requirement is not fulfilled (FAIL).

2.2 Conditions for consent (Art. 7 GDPR)

2.3 User consent

CONSENT_1 `basic`

The messenger MUST obtain user's consent before collecting or processing any personal data.

Preliminary steps

- None;

Verification steps

1. Ask the controller to provide a list of personal data (for which the legal basis for processing is consent) used by the application;
2. For each element of the list, ask the controller to provide information on when it is collected, when consent is requested and what happen if the user does not provide consent;
3. Verify that for each personal data, it is not collected unless consent has been previously obtained;
4. Repeat the previous steps for all possible frontends;

Validation

If the application does not use consent as legal basis for the processing of personal data the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the

controller obtains the user's consent before processing personal data. Otherwise, the requirement is not fulfilled (FAIL).

CONSENT_2 `basic`

The messenger MUST offer the user the possibility to withdraw consent as easily as it is to provide consent.

Preliminary steps

- An account for which consent has been provided;

Verification steps

1. Login with the account;
2. Verify that the option to withdraw consent exists;
3. Proceed to withdraw consent;
4. Verify that the process to withdraw consent is as easy as to provide consent. In order to evaluate the ease of the process, the auditor might consider
   - the presence of a consent plane
   - number of steps to access the relevant option
   - the clarity of the options
5. Repeat the previous steps for all possible frontends;

Validation

If the application does not use consent as legal basis for the processing of personal data the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the auditor comes to the conclusion, that the process to withdraw consent is as easy to use as is the one to give consent. Otherwise, the requirement is not fulfilled (FAIL).

CONSENT_2.a `basic`

The messenger MUST inform the user about practical restrictions regarding the usage of the messenger resulting from consent withdrawal during the withdrawal process.

Prerequisite

- An account for which consent has been provided;

Verification steps

1. Ask the controller to provide a list of restrictions regarding the usage of the messenger resulting from consent withdrawal;
2. Login with the account;
3. Proceed to withdraw consent;
4. Verify if the application informed the user of the practical restrictions regarding the usage of the messenger resulting from consent withdrawal;

5. Repeat the previous steps for all possible frontends;

Validation

If the application does not use consent as legal basis for the processing of personal data the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the application informed the user of the practical restrictions regarding the usage of the messenger resulting from consent withdrawal. Otherwise, the requirement is not fulfilled (FAIL).

CONSENT_3 `basic`

The messenger SHOULD offer a simple dashboard listing all given consent options.

Prerequisite

- An account;

Verification steps

1. Login with the account;
2. Search for a dashboard listing all given consent options;
3. Mark down if a dashboard listing all given consent options exists;
4. Evaluate if the presentation of the dashboard is simple. Indicators for this may include but are not limited to:
   - submenus;
   - font size;
   - color and contrast;
5. Repeat the previous steps for all possible frontends;
6. If a dashboard listing all given consent options does not exist, and the auditor considers it should, ask the controller why;

Validation

If the application does not use consent as legal basis for the processing of personal data the requirement is not applicable (NA). The requirement is fulfilled (PASS) if, for all available frontends, a dashboard listing all given consent options exists AND the auditor considers the presentation of the dashboard to be simple, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CONSENT_3.a `basic`

The dashboard from CONSENT_3 SHOULD list all possible consent options, even those the user did not provide.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all actions that trigger a request for consent within the application;
2. Create an account;
3. Trigger all requests for consent from the list and provide consent to half of the options. Mark down the options that consent has been provided and not provided for;
4. Inspect the content of the dashboard listing the given consent;
5. Verify that all consent marked down above are included in the dashboard, in particular the options that consent has not been provided for;
6. Repeat the previous steps for all possible frontends;
7. If one or more options are not included in the dashboard, and the auditor considers it should, ask the controller why;

Validation

If the application does not use consent as legal basis for the processing of personal data the requirement is not applicable (NA). The requirement is fulfilled (PASS) if, for all available frontends, all options for consent are included in the dashboard OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CONSENT_4 `basic`

The request for consent MUST be written in clear and plain language.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the text files of all requests for consent that may be triggered within the application;
2. Verify if the requests for consent are written in clear and plain language. Indicators for clear and plain language may include but are not limited to;
   - short sentences;
   - active verbs;
   - reader appropriate vocabulary;
3. Repeat the previous steps for all possible frontends;

Validation

If the application does not use consent as legal basis for the processing of personal data the requirement is not applicable (NA). The requirement is fulfilled (PASS) if, for all available frontends, the requests for consent are written in clear and plain language. Otherwise, the requirement is not fulfilled (FAIL).

CONSENT_5 basic

The request for consent MUST be clearly distinguishable from all other matters.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all actions that trigger a request for consent within the application;
2. Create an account;
3. Trigger a request for consent from the list;
4. Mark down if the interface including the request for consent includes any other information or request not related to consent;
5. If the interface includes other information than the request for consent, assess if it is clearly distinguishable from the request for consent. Indicators for distinguishability may include but are not limited to:
   - Order of appearance;
   - Color and contrast;
   - Divisors like dividing or blank lines;
6. Repeat for all requests for consent from the list;
7. Repeat the previous steps for all possible frontends;

Validation

If the application does not use consent as legal basis for the processing of personal data the requirement is not applicable (NA). The requirement is fulfilled (PASS) if, for all available frontends, no interface including a request for consent does include any other information or request not related to consent, OR if the other information within the interface is clearly distinguishable from the request for consent. Otherwise, the requirement is not fulfilled (FAIL).

CONSENT_6 basic

During the process of providing consent, the user MUST be able to access the privacy policy.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all actions that trigger a request for consent within the application;
2. Create an account;
3. Trigger a request for consent from the list;

4. Markdown if the interface including the request for consent includes a link to or a copy of the privacy policy;
5. Repeat for all requests for consent from the list;
6. Repeat the previous steps for all possible frontends;

Validation

If the application does not use consent as legal basis for the processing of personal data the requirement is not applicable (NA). The requirement is fulfilled (PASS) if, for all available frontends, the interface including the request for consent includes a link to or a copy of the privacy policy. Otherwise, the requirement is not fulfilled (FAIL).

CONSENT_7 `basic`

The interface used for requesting consent MUST NOT use deceptive design pattern or any element nudging the user to provide consent.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all actions that trigger a request for consent within the application;
2. Create an account;
3. Trigger a request for consent from the list;
4. Explore the interface of the consent request;
5. Verify that the consent request interface does not include dark patterns. The auditor should consult national guidelines or the EDPB guidelines to assess this compliance [EDP23, Annex][a]. Dark patterns includes, but are not limited to:
   - Overloading
   - Skipping
   - Stirring
   - Hindering
   - Fickle
   - Left in the dark
   - Ambiguous language
   - Misleading color schemes
6. Repeat for all requests for consent from the list;
7. Repeat the previous steps for all possible frontends;

Validation

If the application does not use consent as legal basis for the processing of personal data the requirement is not applicable (NA). The requirement is fulfilled (PASS) if, for all available frontends, no dark pattern or nudging elements were identified in the consent request interfaces. Otherwise, the requirement is not fulfilled (FAIL).

[a] An ontology of dark patterns can be found in [GSBM23]

2.4 Information / transparency (Art. 12/13 GDPR)

2.4.1 Privacy policy

POLICY_1 `basic`

The messenger MUST feature a privacy policy.

Prerequisite

- None;

Verification steps

1. Verify that the messenger application features a privacy policy;
2. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the messenger features a privacy policy. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.a `basic`

The privacy policy MUST include the identity and the contact details of the controller and, where applicable, of the controller's representative.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Verify that the privacy policy includes the identity and the contact details of the controller and, where applicable, of the controller's representative;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the identity and the contact details of the controller and, where applicable, of the controller's representative. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.b `basic`

The privacy policy MUST include the contact details of the data protection officer, where applicable.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Verify that the privacy policy includes the contact details of the data protection officer, where applicable;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the contact details of the data protection officer, where applicable. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.c `basic`

The privacy policy MUST include the purposes of the processing for which the personal data are intended as well as the legal basis for the processing.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Verify that the privacy policy includes the purposes of the processing for which the personal data are intended as well as the legal basis for the processing;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the purposes of the processing for which the personal data are intended as well as the legal basis for the processing. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.d `basic`

If the processing is based on Article 6(1)f GDPR (legitimate interest), the privacy policy MUST include the legitimate interests pursued by the controller or by the third party.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Verify that the privacy policy includes the legitimate interests pursued by the controller or by the third party;
3. Repeat the previous steps for all possible frontends;

Validation

If the processing is not based on legitimate interest, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the legitimate interests pursued by the controller or by the third party. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.e `basic`

The privacy policy MUST include the recipients or categories of recipients of the personal data, if any.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Verify that the privacy policy includes the recipients or categories of recipients of the personal data, if any;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the recipients or categories of recipients of the personal data, if any. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.f `basic`

The privacy policy MUST include where applicable, the fact that the controller intends to transfer personal data to a third country or international organisation and the existence or absence of an adequacy decision by the Commission, or in the case of transfers referred to in Article 46 or 47 GDPR, or the second subparagraph of Article 49(1) GDPR, reference to the appropriate or suitablesafeguards and the means by which to obtain a copy of them or where they have been made available.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Verify that the privacy policy includes where applicable, the fact that the controller intends to transfer personal data to a third country or international organisation and the existence or

absence of an adequacy decision by the Commission, or in the case of transfers referred to in Article 46 or 47 GDPR, or the second subparagraph of Article 49(1) GDPR, reference to the appropriate or suitable safeguards and the means by which to obtain a copy of them or where they have been made available;

3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes where applicable, the fact that the controller intends to transfer personal data to a third country or international organisation and the existence or absence of an adequacy decision by the Commission, or in the case of transfers referred to in Article 46 or 47 GDPR, or the second subparagraph of Article 49(1) GDPR, reference to the appropriate or suitable safeguards and the means by which to obtain a copy of them or where they have been made available. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.g `basic`

The privacy policy MUST include the period for which the personal data will be stored, or if that is not possible, the criteria used to determine that period.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Verify that the privacy policy includes the period for which the personal data will be stored, or if that is not possible, the criteria used to determine that period;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the period for which the personal data will be stored, or if that is not possible, the criteria used to determine that period. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.h `basic`

The privacy policy MUST include the existence of the right to request from the controller access to and rectification or erasure of personal data or restriction of processing concerning the data subject or to object to processing as well as the right to data portability and how to enact these rights.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;

2. Verify that the privacy policy includes the existence of the right to request from the controller access to and rectification or erasure of personal data or restriction of processing concerning the data subject or to object to processing as well as the right to data portability and how to enact these rights;

3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the existence of the right to request from the controller access to and rectification or erasure of personal data or restriction of processing concerning the data subject or to object to processing as well as the right to data portability and how to enact these rights. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.i `basic`

The privacy policy MUST include, where the processing is based on Article 6(1)(a) or 9(2)(a) GDPR, the existence of the right to withdraw consent at any time, without affecting the lawfulness of processing based on consent before its withdrawal.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;

2. Verify that the privacy policy includes the existence of the right to withdraw consent at any time, without affecting the lawfulness of processing based on consent before its withdrawal;

Validation

If the processing is not based on Article 6(1)(a) or 9(2)(a) GDPR, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the existence of the right to withdraw consent at any time, without affecting the lawfulness of processing based on consent before its withdrawal. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.j `basic`

The privacy policy MUST include the right to lodge a complaint with a supervisory authority.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;

2. Verify that the privacy policy includes the right to lodge a complaint with a supervisory authority;

3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the right to lodge a complaint with a supervisory authority. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.k `basic`

The privacy policy MUST include whether the provision of personal data is a statutory or contractual requirement, or a requirement necessary to enter into a contract, as well as whether the data subject is obliged to provide the personal data and of the possible consequences of failure to provide such data.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Verify that the privacy policy includes whether the provision of personal data is a statutory or contractual requirement, or a requirement necessary to enter into a contract, as well as whether the data subject is obliged to provide the personal data and of the possible consequences of failure to provide such data;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes whether the provision of personal data is a statutory or contractual requirement, or a requirement necessary to enter into a contract, as well as whether the data subject is obliged to provide the personal data and of the possible consequences of failure to provide such data. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.l `basic`

The privacy policy MUST include the existence of automated decision-making, including profiling, referred to in Article 22(1) and (4) GDPR and, in those cases, meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject.

Prerequisites

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Verify that the privacy policy includes the existence of automated decision-making, including profiling, referred to in Article 22(1) and (4) and, at least in those cases, meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the existence of automated decision-making, including profiling, referred to in Article 22(1) and (4) and, at least in those cases, meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_1.m `basic`

The privacy policy SHOULD include the name and contact info of the competent supervisory authority.

Prerequisites

- The privacy policy;

1. Inspect the privacy policy;
2. Verify that the privacy policy includes the correct name and contact info of the competent supervisory authority;
3. If the privacy policy does not include the correct name and contact info of the competent supervisory authority and the auditor considers it should be, ask the controller why;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy includes the correct name and contact info of the competent supervisory authority OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_2 `basic`

The privacy policy SHOULD be accessible before to the user starts the account creation process.

Prerequisite

- None;

Verification steps

1. Verify that the privacy policy is accessible within the application without starting the account creation process;
2. If the privacy policy is not accessible without starting the account creation process and the auditor considers it should be, ask the controller why;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy is accessible without starting the account creation process OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_3 basic

The privacy policy MUST be displayed to the user upon account creation or upon collection of personal data from the user, whichever comes first.

Prerequisite

- None;

Verification steps

1. Ask the controller for a list of situations in which personal data is collected before the account creation process;
2. Enact a situation from the list;
3. Mark down if the privacy policy is displayed in this situation;
4. Repeat the previous steps for all given situations;
5. Proceed to the account creation process;
6. Mark down if the privacy policy is displayed during the creation process;
7. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, and for each situation where personal data is collected before the account creation process, the privacy policy is displayed OR the privacy policy is displayed during the creation process if there are no situations in which personal data is collected beforehand. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_4 basic

The information provided in the privacy policy MAY be provided in combination with standardised icons in order to give in an easily visible, intelligible and clearly legible manner a meaningful overview of the intended processing.

Prerequisite

- The privacy policy;

Verification steps

1. Inspect the privacy policy;
2. Mark down if the used icons in the privacy policy are easily visible, intelligible, clearly legible and provide a meaningful overview of the intended processing;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the auditor comes to the conclusion that the used icons in the privacy policy are easily visible, intelligible, legible and sufficiently meaningful and provide an overview of the intended processing. Otherwise, the requirement is not fulfilled (FAIL).

POLICY_5 `basic`

The privacy policy MUST be written in the official language or languages of the country in that the service is provided.

Prerequisite

- None;

Verification steps

1. Mark down the official languages of the country in that the service is provided and currently audited;
2. Verify that the privacy policy is provided in the official languages marked down in the previous step;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy policy is written in every official language of the country in which the service is provided. Otherwise, the requirement is not fulfilled (FAIL).

2.5 Right to access (Art. 15 GDPR)

ACC_RIGHT_1 `basic`

The controller MUST provide a way for the user to obtain a copy of personal data concerning the user.

Prerequisite

- A populated account A;

Verification steps

1. Mark down all provided means of the controller by which the user can request a copy of personal data concerning the user;
2. Request a copy of personal data through one of the means identified in the previous step;
3. Mark down if the access right request has been answered;
4. Inspect the content of the data provided by the controller;
5. Verify that the data provided by the controller is a plausible copy of all personal data concerning the user A;
6. Repeat the previous steps for all identified means from the first step;

Validation

The requirement is fulfilled (PASS) if, at least one process to request a copy of personal data concerning the user exists AND the data provided by the controller in response to such a request through any of the available means is a plausible copy of all personal data concerning the user. Otherwise, the requirement is not fulfilled (FAIL).

ACC_RIGHT_2 `basic`

If the request to access was made by electronic means, the controller MUST provide the copy in a commonly used electronic form.

Prerequisite

- A populated account A;

Verification steps

1. Mark down all available options in which the request to provide a copy of personal data processed by the controller can be made through electronical means;
2. Request a copy of personal data through one of the means identified in the previous step;
3. Verify that the copy is provided in a commonly used electronic form. Commonly used electronic forms includes, but are not limited to: PDF, TXT, ODT, HTML;
4. Repeat the previous steps for all identified means of the first step;

Validation

If the access request cannot be made by electronical means the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, all requests are answered AND the received copy was provided in a commonly used electronic format. Otherwise, the requirement is not fulfilled (FAIL).

ACC_RIGHT_3 `basic`

The controller SHOULD NOT provide the copy in a proprietary format.

Prerequisite

- A populated account;

Verification steps

1. Mark down all provided means of the controller by which the user can request a copy of personal data concerning the user;
2. Request a copy of personal data through one of the means identified in the previous step;
3. Mark down if the copy is provided in a proprietary format. Typical non-proprietary formats include, but are not limited to: TXT, ODT, HTML;
4. If the copy is provided in a proprietary format, and the auditor considers it should not be, ask the controller why;
5. Repeat the previous steps for all identified means of the first step;

Validation

The requirement is fulfilled (PASS) if, the received copies are provided in a non-proprietary format OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ACC_RIGHT_4 `basic`

The controller MUST provide at least one copy in an human readable format.

Prerequisite

- A populated account;

Verification steps

1. Mark down all provided means of the controller by which the user can request a copy of personal data concerning the user;
2. Request a copy of personal data through one of the means identified in the previous step;
3. Mark down if the answer includes a copy in an human readable format;
4. Repeat the previous steps for all identified means of the first step;

Validation

The requirement is fulfilled (PASS) if, the received answer includes a copy that is in a human readable format. Otherwise, the requirement is not fulfilled (FAIL).

ACC_RIGHT_5 `basic`

The controller MAY provide the copy in a machine-readable format in addition to a human readable format.

Prerequisite

- A populated account;

Verification steps

1. Mark down all provided means of the controller by which the user can request a copy of personal data concerning the user;
2. Request a copy of personal data through one of the means identified in the previous step;
3. Mark down if the answer includes a copy in a machine-readable format. Commonly used machine readable formats include, but are not limited to: CSV, JSON, XML, YAML, TOML;
4. Repeat the previous steps for all identified means of the first step;

Validation

If the provided answer to the access request does not include an additional copy indicating a machine-readable copy the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, the received additional copy is in a machine readable format.Otherwise, the requirement is not fulfilled (FAIL).

ACC_RIGHT_6 `basic`

The controller SHOULD provide the option to download the copy within the application.

Prerequisite

- A populated account;

Verification steps

1. Mark down all provided means of the controller by which the user can request a copy of personal data concerning the user;
2. Request a copy of personal data through one of the means identified in the previous step;
3. Mark down if the controller provides an option to download the requested data within the application;
4. Repeat the previous step for all possible frontends;
5. Repeat all the previous steps for all identified means of the first step;
6. If the option to download the copy within the application is not provided, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, an option to download the copy within the all possible frontends of the application is provided OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ACC_RIGHT_7 `basic`

The controller SHOULD NOT request further personal data beyond what is absolutely necessary in order to fulfill the request to a copy of personal data.

Prerequisite

- A populated account;

Verification steps

1. Mark down all provided means of the controller by which the user can request a copy of personal data concerning the user;
2. Ask the controller for a list of personal data points that the controller requires from the user in oder to process the access right request for each identified mean of the previous step;
3. If the auditor considers the controller requests further personal data beyond what is absolutely necessary in order to fulfill the request, ask the controller why;. Typical data points that might indicate overreach of the controller include, but are not limited to:
   - Any data not already present by the controller;
   - Unredacted copies of identity cards or other identifying documents;
   - Authentication through a third party;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller does not request further personal data beyond what is absolutely necessary in order to fulfill the request OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ACC_RIGHT_8 `basic`

The controller SHOULD identify the user through an authentication mechanism such as the same credentials used by the user to log-in to the messenger service.

Prerequisite

- A populated account;

Verification steps

1. Mark down all provided means of the controller by which the user can request a copy of personal data concerning the user;
2. Proceed with on of the means identified in the previous step to enact an access right request;
3. Mark down if during the access right request, the user is identified through an authentication mechanism such as the same credentials used by the user to log-in to the messenger service;
4. Repeat the previous steps for all identified means of the first step;
5. If the controller does not identify the user through an authentication mechanism such as the same credentials used by the user to log-in to the messenger service, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all available means perform an access right request the user is identified through an authentication mechanism such as the same credentials used by the user to log-in to the messenger service OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.6 Right to rectification (Art. 16 GDPR)

REC_RIGHT_1 `basic`

The controller SHOULD provide a way for the user to rectify inaccurate personal data concerning him or her.

Prerequisite

- A populated account.

Verification steps

1. Login with the account;
2. Verify that the controller provides a way for the user to rectify inaccurate personal data concerning them;
3. For each personal data concerning the user proceed to a rectification of the personal data using a different plausible value;
4. Verify that the rectification of each personal data was successful;
5. Repeat the previous steps for all possible frontends;
6. If the controller does not provide a way for the user to rectify inaccurate personal data concerning the user, and the auditor considers it should, ask the controller why;
7. If one or more personal data can not be rectified, and the auditor considers they should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller provides a way for the user to rectify inaccurate personal data concerning the user AND all personal data concerning the user was successfully rectified, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

REC_RIGHT_2 `basic`

The controller SHOULD place the mechanism to start the process to rectify inaccurate personal data concerning the user adjacent to the user profile.

Prerequisite

- A populated account.

Verification steps

1. Login with the account;
2. Verify that the mechanism to start the process to rectify inaccurate personal data concerning the user is adjacent to the user profile;
3. Repeat the previous steps for all possible frontends;
4. If the mechanism to start the process to rectify inaccurate personal data concerning the user is not adjacent to the user profile, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the mechanism to start the process to rectify inaccurate personal data concerning the user is adjacent to the user profile OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

REC_RIGHT_3 `basic`

The controller MAY offer a feature to edit messages sent by the user.

Prerequisite

- A populated account A for each available frontend;
- An account B that is a contact of A;

Verification steps

1. Login to the account A;
2. Select one unedited message m in each discussion and edit them to message m';
3. Verify that each selected message is successfully edited;
4. Login to the account B;
5. Verify that the message m has changed to the message m';
6. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature editing sent messages, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the message was successfully edited and displayed for A and B. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If a feature to edit messages sent by the user is not offered , the following requirement is not applicable: REC_RIGHT_3.a

REC_RIGHT_3.a basic

An edited message MUST be clearly indicated as such to all recipients and the sender of the message.

Prerequisite

- A populated account A for each available frontend;
- An account B that is a contact of A and member of the same group as A.

Verification steps

1. Login with the account A;
2. Select an unedited message m from the conversation with B and edit it to message m';
3. Select an unedited message m_g from a group G that B is part of and edit it to message m'_g;
4. Login with account B;
5. Mark down if the received message m' is clearly indicated as edited;
6. Mark down if the received message m'_g is clearly indicated as edited;
7. Repeat the previous steps for all possible frontends;

Validation

If a feature to edit messages sent by the user is not offered the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, all edited messages are clearly indicated as such. Otherwise, the requirement is not fulfilled (FAIL).

2.7 Right to erasure (Art. 17 GDPR)

2.7.1 Account deletion

ACC_DEL_1 basic

The messenger application SHOULD offer a mean to delete the account.

Prerequisite

- A populated account for each available frontend;
- A account T that is contact of the populated account;

Verification steps

1. Login to the account A_n;
2. Verify that the messenger application features an option to delete the account;
3. Proceed to the account deletion process;
4. Try to login with the credential of the account;
5. Verify that it is not possible to login;

6. Verify, using the account T, that the account from step one is not visible. To assess the *visibility* of the account, the auditor can, but is not limited to, inspect the contact list of connected account and use the contact search method of the messenger;
7. Repeat the previous steps for all possible frontends;
8. If one or all of the user accessible frontends don't offer a way to delete the account and the auditor considers one or more tested frontends ought to offer such a mean, ask the controller why;
9. If the auditor successfully logged in after the account was deleted, and the auditor consider it should not be the case, ask the controller why;
10. If the account is visible after it was deleted, and the auditor considers it should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, a mean to delete the account is offered AND the login failed AND the none of the deleted accounts A_n was not visible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the account deletion feature is not provided but this requirement is fulfilled (PASS), the following requirements are not-applicable (NA): ACC_DEL_1.a, ACC_DEL_2, ACC_DEL_2.a, ACC_DEL_2.b, ACC_DEL_2.c, ACC_DEL_3, ACC_DEL_3.a, ACC_DEL_4, ACC_DEL_5.

ACC_DEL_1.a `basic`

The process to delete the account MUST be as easily accessible as the account creation process.

Prerequisite

- A populated account for each available frontend;

Verification steps

1. Login to the account;
2. Proceed to account deletion;
3. Verify that the account deletion process is as easily accessible as the account creation process. To assess the accessibility, the auditor can consider, but is not limited to, the layout, appropriate headings and paragraphing [a]
4. Repeat the previous steps for all possible frontends;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, **the auditor considers** that account deletion process is as easily accessible as the account creation process. Otherwise, the requirement is not fulfilled (FAIL).

[a] See paragraph 140 of EDPB Guidelines 01/2022 on data subject rights - Right of access [EDP22, par.140]

ACC_DEL_2 `basic`

Upon activation of the account deletion feature, the deletion of data SHOULD be executed without undue delay.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the details of the account deletion process;
2. Verify that upon receiving a request for account deletion and starting the deletion, no more than 72 hours pass;
3. Repeat the previous steps for all possible frontends;
4. If the timespan exceeds 72 hours and the and the auditor considers it should not be, ask the controller why;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the data is deleted within 72 hours, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If this requirement is fulfilled, the following requirements are not-applicable (NA): ACC_DEL_2.a, ACC_DEL_2.b, ACC_DEL_2.c.

ACC_DEL_2.a `basic`

The messenger MAY feature a retention period no longer than 4 weeks.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the duration of the retention period;
2. Verify that the retention period is not longer than 4 weeks;
3. Repeat the previous steps for all possible frontends;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled, OR if the account deletion process is executed without delay, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the retention period is not longer than 4 weeks, if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ACC_DEL_2.b `basic`

The user MUST be informed of the retention period and its duration upon performing an account deletion request.

Prerequisite

- A populated account for each available frontend;

Verification steps

1. Login to the account;
2. Proceed to account deletion;
3. Verify that the information on the retention period and its duration are provided to the user;
4. Repeat the previous steps for all possible frontends;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled, OR if the account deletion process is executed without delay, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, information on the retention period and its duration are provided to the user upon performing an account deletion request. Otherwise, the requirement is not fulfilled (FAIL).

ACC_DEL_2.c `basic`

The messenger MUST feature a process to trigger the deletion without further delay if the user wishes to waiver the retention feature.

Prerequisite

- A populated account for each available frontend;

Verification steps

1. Login to the account;
2. Proceed to account deletion;
3. Verify that it is possible to trigger the deletion without further delay;
4. Repeat the previous steps for all possible frontends;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled, OR if the account deletion process is executed without delay, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, it is possible to trigger the deletion without further delay. Otherwise, the requirement is not fulfilled (FAIL).

ACC_DEL_3 `basic`

Upon account deletion all personal data SHOULD be deleted from the user's device.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all personal data stored on the device;
2. Ask the controller to provide the details of the account deletion process on the device;
3. Verify that all personal data stored on the device is deleted upon account deletion;
4. Repeat the previous steps for all possible frontends;
5. If one or more personal are not deleted from the device upon account deletion, ask the controller why;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, all personal data stored on the device is deleted upon account deletion OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ACC_DEL_4 `basic`

Upon account deletion, the messages and content authored by the user SHOULD be modified to change the author name to a default value.

Prerequisite

- A populated account;

Verification steps

1. Login to the account;
2. Proceed to account deletion;
3. Verify, with other accounts involved in discussions with the deleted account, that each message and content authored by the deleted account has its author name set to a default value;
4. Repeat the previous steps for all possible frontends;
5. If one or more messages or content authored by the deleted account has not its author name set to a default value, and the auditor considers it should be, ask the controller why;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, all messages and content authored by the deleted account have their author name set to a default value, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ACC_DEL_5 `basic`

During the account deletion process the application SHOULD provide the user an easily accessible way to perform a backup as in described in 2.11.18.

Prerequisite

- A populated account for each available frontend;

Verification steps

1. Login to the account;
2. Proceed to account deletion;
3. Verify that, during the account deletion process, the user is offered a way to perform a backup as described in 2.11.18 of the data stored on the device;
4. Verify that the option to perform the backup is easily accessible;
5. Repeat the previous steps for all possible frontends;
6. If the user is not offered a way to perform a backup of the data stored on the device and the auditor considers it should, ask the controller why;
7. If the option to perform the backup is not easily accessible, and the auditor considers it should, ask the controller why;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user is offered a way to perform a backup of the data stored on the device AND the option to perform the backup is easily accessible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ACC_DEL_6 `basic`

During the account deletion process the application SHOULD provide the user an easily accessible way to perform a data export as described in [fIS24, Sec. Data export] in order to export all personal data in a structured, commonly used and machine-readable format.

Prerequisite

- A populated account for each available frontend;

Verification steps

1. Login to the account;
2. Proceed to account deletion;
3. Verify that, during the account deletion process, the user is offered a way to perform an export of all personal data provided by the user to the controller as described in 2.8.1;
4. Verify that the option to perform the data export is easily accessible;

5. Repeat the previous steps for all possible frontends;
6. If the user is not offered a way to export all personal data provided to the controller, and the auditor considers it should, ask the controller why;
7. If the option to perform the data export is not easily accessible, and the auditor considers it should be, ask the controller why;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user is offered a way to export all personal data provided to the controller as described in 2.8.1 AND the option to perform the data export is easily accessible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ACC_DEL_6.a `basic`

During the account deletion process the application SHOULD provide the user an easily accessible way to perform a direct data transfer to another controller as described in [fIS24, Sec. Direct data transfer].

Prerequisite

- A populated account for each available frontend;

Verification steps

1. Login to the account;
2. Proceed to account deletion;
3. Verify that, during the account deletion process, the user is offered a way to directly transmit a data export to another controller as described in 2.8.3;
4. Verify that the option to perform the direct data export is easily accessible;
5. Repeat the previous steps for all possible frontends;
6. If the user is not offered a way to perform a direct data export to another controller, and the auditor considers it should, ask the controller why;.
7. If the option to perform the data export is not easily accessible, and the auditor considers it should be, ask the controller why;

Validation

If the account deletion feature is not available AND ACC_DEL_1 is fulfilled, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user is offered a way to perform a direct data export to another controller as described in 2.8.3 AND the option to perform the direct data export is easily accessible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.7.2 Message deletion

MSG_DEL_1 `basic`

The messenger SHOULD feature an option to delete a message including its metadata authored by the user.

Prerequisite

- A populated account for each available frontend;;

Verification steps

1. Login with the account;
2. Mark down if for each message type the messenger offers and authored by the user, an option to delete the message exists;
3. In each conversation, select one text message and one content message and delete them;
4. Verify that, from the point of view of the account, each selected message and its metadata is removed from the conversation;
5. Repeat the previous steps for all possible frontends;
6. If the option to delete a message does not exist and the auditor considers it should beask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the option to delete a message does exist AND all selected messages and their metadata are deleted, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not feature an option to delete a message AND this requirement is fulfilled, the following requirements are not applicable (NA): MSG_DEL_1.a

MSG_DEL_1.a `basic`

The message deletion feature MAY delete the message for all participants of the conversation in addition to the user.

Prerequisite

- A populated account for each available frontend;, A;
- Accounts that are contact of A, $\mathcal{B} = \{B_1, B_2 \dots B_n\}$

Verification steps

1. Login with the account A;
2. In each available conversation, select one text message and one content message authored by the user and delete them;

3. For account B_i contact of A, login with B_i and inspect the messages of A;
4. Verify, that each selected message is deleted from the point of view of all accounts B_i.
5. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature an option to delete a message AND MSG_DEL_1 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, all selected messages are deleted for all tested accounts B_i, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

MSG_DEL_2 `basic`

A deleted message MUST be clearly indicated to all participants of the conversation.

Prerequisite

- A populated account for each available frontend;, A;
- Accounts that are contact of A, $\mathcal{B} = \{B_1, B_2 \dots B_n\}$

Verification steps

1. Login with the account A;
2. In each available conversation, select one text message and one content message and delete them;
3. For account B_i contact of A, login with B_i and inspect the messages of A;
4. Verify, that each selected message is clearly indicated as deleted from the point of view of all accounts B_i;
5. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature an option to delete a message AND MSG_DEL_1 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, all selected messages are clearly indicated as deleted from the point of view of all tested accounts B_i. Otherwise, the requirement is not fulfilled (FAIL).

MSG_DEL_3 `basic`

A self deleting message MUST be clearly indicated to all participants of the conversation.

Prerequisite

- A populated account for each available frontend;, A;
- Accounts that are contact of A, $\mathcal{B} = \{B_1, B_2 \dots B_n\}$;

Verification steps

1. Login with the account A;
2. In each available conversation, send one self deleting message;
3. For account B_i contact of A, login with B_i and inspect the messages of A;
4. Verify, that each self deleting message is indicated as such from all accounts B_i;
5. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature self deleting messages, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, for all accounts, all self deleting messages are clearly indicated as such. Otherwise, the requirement is not fulfilled (FAIL).

MSG_DEL_3.a `basic`

The remaining time of self deleting message MUST be clearly indicated to all participants of the conversation.

Prerequisite

- A populated account for each available frontend;, A;
- Accounts that are contact of A, $\mathcal{B} = \{B_1, B_2 \dots B_n\}$;

Verification steps

1. Login with the account A;
2. In each conversation, send one self deleting message;
3. For account B_i contact of A, login with B_i and inspect the messages of A;
4. Verify, that the remaining time of all self deleting messages is clearly indicated as such from all accounts B_i;
5. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature self deleting messages, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, for all accounts, the remaining time of all self deleting messages is clearly indicated. Otherwise, the requirement is not fulfilled (FAIL).

2.7.3 Data deletion

DATA_DEL_1 `basic`

The messenger application SHOULD feature an option to delete all data associated with the messenger application from the device.

- A populated account for each available frontend;

Verification steps

1. Login with the account;
2. Mark down if the application features an option to delete all data associated with the messenger application from the device;
3. Ask the controller to provide the list of data associated with the messenger application stored on the device;
4. Ask the controller to describe how the ”delete all data” process is handeld by the application;
5. Verify that each data associated with the messenger application stored on the device is deleted during the ”delete all data” process;
6. Repeat the previous steps for all possible frontends;
7. If the application does not feature an option to delete all data associated with the messenger application from the device, and the auditor considers it should, ask the controller why;.
8. If one or more data are not deleted by the ”delete all data” process, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application features an option to delete all data associated with the messenger application from the device AND all data associated with the messenger application stored on the device is deleted during the ”delete all data” process, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

DATA_DEL_2 `basic`

Upon uninstalling the application, all associated data stored on the device SHOULD be deleted.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all data associated to the application stored on the device;
2. Ask the controller to provide information on the actions taken on the device upon uninstalling the application;
3. Verify that all data associated with the application stored on the device is deleted upon uninstalling the application;
4. Repeat the previous steps for all possible frontends;
5. If one or more data associated to the application stored on the device are not deleted upon uninstalling the application, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all data associated with the application stored on the device are deleted upon uninstalling the application OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.8 Data portability (Art. 20 GDPR)

2.8.1 Data export

PORT_1 `basic`

The application SHOULD feature a functionality to export personal data concerning the user, which they have provided to a controller, in a structured, commonly used and machine-readable format.

Prerequisite

- A populated account for each available frontend;;

Verification steps

1. Using one of the Accounts A, verify that a there is a functionality to export personal data concerning the user, which was provided to a controller. The auditor should check at least the Account preference panel, the Website, FAQ and the privacy policy;
2. Proceed to the export procedure;
3. Open and inspect the retrieved data;
4. Verify that the retrieved data plausibly corresponds to the data provided by the user to the controller. Elements to be verified include, but are not limited to: data provided during the account creation (username, email, phone number...), profile information (bio, picture...), messages sent by the user...
5. Verify that the retrieved data is in a structured, commonly used and machine-readable format. Structured, commonly used and machine-readable format includes, but are not limited to: CSV, JSON, XML, YAML, TOML; and is provided along with metadata at the best possible level of granularity[a];
6. Repeat the previous steps for all possible frontends;
7. If a functionality to export personal data concerning the user, which was provided to a controller, in a structured, commonly used and machine-readable format does not exist, and the auditor considers it should, ask the controller why;.
8. If the retrieved data does not plausibly corresponds to the data provided by the user to the controller, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, a functionality to export personal data concerning the user, which was provided to the controller, in a structured, commonly used and machine-readable format exists AND the retrieved data plausibly corresponds to the data provided by the user to the controller AND the retrieved data is in a structured, commonly used and machine-readable format, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not feature a functionality to export personal data concerning the user AND this requirement is fulfilled, the following requirements are not applicable: PORT_2, PORT_3, PORT_4, PORT_5, PORT_6.

[a] See [WP217]

PORT_2 `basic`

The specification of the format used for the data export MUST be freely available.

Prerequisite

- None;

Verification steps

1. Identify the format used for the data export;
2. Mark down if the specification used as the data export format is freely available. The auditor should include reasonable sources of the developer of the specification or the controller itself;

Validation

If a functionality to export personal data concerning the user does not exist AND PORT_1 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the specification of the format used for the data export is freely available. Otherwise, the requirement is not fulfilled (FAIL).

PORT_3 `basic`

The data exported SHOULD NOT include the contact list of the user.

Prerequisite

- A populated account

Verification steps

1. Login with the account A;
2. Proceed to the data export procedure;
3. For each of the contact of the account A, search for each identifier (email, phone number, pseudonym...) of the contact in the data export;
4. Mark down if one of the contacts of A were identified in the data export; Repeat the previous steps for all possible frontends;
5. If the data export included any contact data and the auditor considers it should not be, ask the controller why;

Validation

If a functionality to export personal data concerning the user does not exist AND PORT_1 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all

available frontends, the data export does not include contacts of the user, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PORT_4 `basic`

The data exported SHOULD include all messages authored by the user.

Prerequisite

- A populated account

Verification steps

1. Login with the account A;
2. Prepare a list of at least ten messages authored by the user. The list should include at least two messages of each message functionality such as direct, group and broadcast messages and media types such as images or videos that the messenger offers;
3. Proceed to the data export procedure;
4. Mark down if each message listed in step two is included in the data export; Repeat the previous steps for all possible frontends;
5. If one or more messages listed in step two are not included in the data export, and the auditor considers they should be, ask the controller why;.

Validation

If a functionality to export personal data concerning the user does not exist AND PORT_1 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the data export includes all messages listed, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PORT_5 `basic`

The data exported SHOULD NOT include messages not authored by the user.

Prerequisite

- A populated account

Verification steps

1. Login with the account A;
2. Prepare a list of at least ten messages received by the user. The list should include at least two messages of each message functionality such as direct, group and broadcast messages that the messenger offers;
3. Proceed to the data export procedure;
4. Mark down if any message listed in step two is included in the data export; Repeat the previous steps for all possible frontends;

5. If one or more messages listed in step two are included in the data export, and the auditor considers they should not be, ask the controller why;.

Validation

If a functionality to export personal data concerning the user does not exist AND PORT_1 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the data export does not include any message listed, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.8.2 Data import

PORT_7 `basic`

The messenger SHOULD feature a functionality to import personal data in a structured, commonly used and machine-readable format.

Prerequisite

- Data export D_int in a structured, commonly used and machine-readable format obtained from the audited messenger service;
- Data export D_ext in a structured, commonly used and machine-readable format obtained from another messenger service;

Verification steps

1. Create a new account A_int;
2. Verify that the messenger features a functionality to import personal data in a structured, commonly used and machine-readable format;
3. Proceed to import the data export D_int (obtained from the audited messenger);
4. Mark down if the data import is successful by verifying that the process is completed and the data was correctly imported without errors;
5. Create a new account A_ext;
6. Proceed to import the data export D_ext (obtained from another messenger);
7. Mark down if the data import is successful by verifying that the process is completed and the data was correctly imported without errors;
8. Repeat the previous steps for all possible frontends;
9. If a functionality to import personal data in a structured, commonly used and machine-readable format does not exist, and the auditor considers it should, ask the controller why;
10. If one or more datapoints from the data export was not imported and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, a functionality to import personal data in a structured, commonly used and machine-readable format exists AND all data import were successful, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent step

If the messenger does not feature a functionality to import personal data AND this requirement is fulfilled, the following requirements are not applicable (NA): PORT_7.a.

PORT_7.a `basic`

The functionality to import personal data MUST be offered during the account creation process.

Prerequisite

- None;

Verification steps

1. Create an account;
2. Verify that the functionality to import personal data was offered during the account creation process;
3. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature a functionality to import personal data AND PORT_7 is fulfilled, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the functionality to import personal data was offered during the account creation process. Otherwise, the requirement is not fulfilled (FAIL).

2.8.3 Direct data transfer

PORT_8 `basic`

The messenger SHOULD feature a functionality to directly transmit a data export to another controller.

Prerequisite

- A populated account A;

Verification steps

1. Login with the account A and search for the functionality to directly transmit a data export to another controller;
2. Mark down if the functionality to directly transmit a data export to another controller exists;
3. Ask the controller on details on how the direct data transfer to another controller works;

4. Repeat the previous steps for all possible frontends;
5. If the functionality to directly transmit a data export to another controller does not exist, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the functionality to directly transmit a data export to another controller exists , OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PORT_9 `basic`

The messenger SHOULD feature a functionality to directly receive a data export from another controller.

Prerequisite

- An account A;

Verification steps

1. Login with the account A and search for the functionality to directly receive a data export from another controller;
2. Mark down if the functionality to directly receive a data export from another controller exists;
3. Ask the controller on details on how the direct data trasfere to another controller works;
4. Repeat the previous steps for all possible frontends;
5. If the functionality to directly receive a data export from another controller does not exist, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the functionality to directly receive a data export from another controller exists , OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9 Data protection by design and by default (Art. 25 GDPR)

DAT_PROTEC_1 `basic`

Personal data SHOULD be deleted or anonymised as soon as possible.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of personal data used by the application, along with their

management process;

2. Verify that each personal data is either erased or anonymised as soon as possible;
3. Repeat the previous steps for all possible frontends;
4. If the auditor considers the erasure or anonymisation of a personal data point is to late or should occur sooner, ask the controller for a statement on why the datapoints are deleted or anonymised only at this point in time and not sooner;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, every personal data used by the application is either erased or anonymised as soon as possible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

DAT_PROTEC_2 `basic`

Personal data SHOULD pseudonymised as soon as possible.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all personal data used by the application, along with their management process;
2. Verify that each personal data is pseudonymised as soon as possible;
3. Repeat the previous steps for all possible frontends;
4. If the auditor considers the pseudonymisation a personal data point is to late or should occur sooner, ask the controller for a statement on why the datapoints are only pseudonymised at this point in time and not sooner;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, every personal data used by the application is pseudonymised as soon as possible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

DAT_PROTEC_3 `basic`

Anonymisation SHOULD be performed according to the state of the art.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of personal data that is anonymised and the corresponding techniques used for the anonymisation;

2. Verify that anonymisation techniques are compliant with the state of the art and current best practices. The auditor can consult national guidelines or resources such as the following to assess this compliance:
    - WP29 Opinion 05/2014 on Anonymisation Techniques [WP214a]
3. Repeat the previous steps for all possible frontends;
4. If one or more anonymisation techniques are not compliant with the state of the art and current best practices or not fitting for the set of personal data that is to be anonymised and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all anonymisation techniques are compliant with the state of the art and current best practices and are fitting for the set of personal data that is to be anonymised, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

DAT_PROTEC_4 `basic`

Pseudonymisation SHOULD be performed according to the state of the art.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of personal data that is pseudonymised and the corresponding techniques used for the pseudonymisation;
2. Verify that pseudonymisation techniques are compliant with the state of the art and current best practices. The auditor can consult national guidelines or resources such as the following to assess this compliance:
    - WP29 Opinion 05/2014 on Anonymisation Techniques [WP214a]
3. Repeat the previous steps for all possible frontends;
4. If one or more pseudonymisation techniques are not compliant with the state of the art and current best practices or not fitting for the set of personal data that is to be pseudonymised and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all pseudonymisation techniques are compliant with the state of the art and current best practices and are fitting for the set of personal data that is to be pseudonymised, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.1 Identifiers

IDENTIFIER_1 `basic`

The messenger application SHOULD NOT use persistent identifiers of the device.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all identifiers used by the application;
2. Verify that none of the identifiers used by the application is a persistent identifier of the device. Persistent identifiers includes but are not limited to:
   - Device/equipment identifiers: Device ID, UDID IMEI, MEID,
   - Sim card identifiers: SIM Serial Number, Suscriber ID, ICC ID
   - Network Interface identifier: MAC address
3. Repeat the previous steps for all possible frontends;
4. If one or more persistent identifiers are used and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no persistent device identifier is used, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

IDENTIFIER_1 `intermediate`

The messenger application SHOULD NOT use persistent identifiers of the device.

Prerequisite

- Access to the source code of the application for all available frontends

Verification steps

1. Open the source code of the application;
2. Search the source code for indicators that suggest that the application is directly accessing persistent identifiers of the device. The auditor can search for system calls or API functions known to provide access to persistent identifiers of the device. Such system calls or API functions include, but are not limited to:
   - Android:
     - WifiInfo#getMacAddress
     - TelephonyManager#getDeviceId
     - TelephonyManager#getImei
     - TelephonyManager#getMeid

- TelephonyManager#getSimSerialNumber
- TelephonyManager#getSubscriberId
- Build#getSerial

- Windows:
  - GetAdaptersAddresses
  - GetPhysicalAddress

- iOS:

3. Search the source code for indicators that suggest that the application is indirectly accessing persistent identifiers of the device. The auditor can search for access to resources that may include persistent identifiers. Such resources include, but are not limited to:

   - Android/Linux:
     - /proc/net/arp
     - /sys/class/net/XXX/address
     - /var/log/syslog
     - /etc/NetworkManager/system-connections/
     - ioctl: SIOCGIFHWADDR
   - Windows:
     - Registry: bimacaddres_h and bimacaddres_l

4. Repeat the previous steps for all possible frontends;
5. If one or more persistent identifiers of the device are accessed by the application, and the auditor considers they should not be, ask the controller why;
6. If one or more resources that may include persistent identifiers are accessed by the application and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no persistent identifiers of the device is accessed by the application AND no resources that may include persistent identifiers is accessed by the application, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

IDENTIFIER_2 `basic`

The messenger SHOULD NOT use identifiers derived from the technical characteristics of the device. (E.g. fingerprinting)

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all identifiers used by the application, along with how they are created;
2. Verify that none of the identifiers is derived from the technical characteristics of the device. To

assess if an identifier is derived from the technical characteristics of the device, the auditor can consult national guidelines or resources such as the following:

- WP29 Opinion 9/2014 on the application of Directive 2002/58/EC to device fingerprinting [WP214b]

3. Repeat the previous steps for all possible frontends;
4. If one or more identifiers are derived from the technical characteristics of the device and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no identifier used by the application is derived from the technical characteristics of the device , OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

IDENTIFIER_2 `intermediate`

The messenger SHOULD NOT use identifiers derived from the technical characteristics of the device. (E.g. fingerprinting)

Web `intermediate`

Prerequisites

- A populated account
- A fingerprinting detection tool installed in a browser;
  - FPMON [a]

Verification steps

1. Activate the fingerprinting detection tool and monitor its results while performing the following actions:
   (a) Visit the login page of the service;
   (b) Login to the service;
   (c) Proceed to a standard use procedure (A.1.5);
2. Verify that no fingerprinting method is present in the application. To assess the presence of a fingerprinting mechanism, the auditor can rely on the score provided by the application as well as the origin of the flagged elements;
3. Repeat the previous steps for all possible web frontends;
4. If one or more fingerprinting methods are present in the application, and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no fingerprinting method is present in the application, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

[a] https://fpmon.github.io/fingerprinting-monitor/

Mobile `intermediate`

Prerequisites

- Access to the source code;

Verification steps

1. Inspect the source code;
2. Retrieve the list of libraries included in the application;
3. Verify that none of these libraries is known to include a device fingerprinting mechanism. Libraries known to include a device fingerprinting mechanism include, but are not limited to:
   - fingerprintjs [a]
   - ipqualityscore [b]
   - Chartboost
   - Tapjoy
   - Kochava
   - Amazon Mobile Ads
   - ThreatMetrix
   - INFOnline
   - Iovation
   - Kontagent
4. Repeat the previous steps for all possible application frontends;
5. If one or more libraries included in the application are known to include a device fingerprinting mechanism, and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no libraries included in the application is known to include a device fingerprinting mechanism, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

[a] https://github.com/fingerprintjs/
[b] https://www.ipqualityscore.com/documentation/mobile-device-fingerprinting-sdk/android-ios-install

IDENTIFIER_3 `basic`

The messenger application SHOULD use identifiers that are specific to the application.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all identifiers used by the application;
2. Verify that each identifier of the list is specific to the application. Such identifiers include, but are not limited to:
   - Android: Firebase installation ID (FID), globally-unique ID (GUID), App set ID, Advertising ID;
   - iOS: Vendor ID, Advertising ID;
   - Windows: Advertising ID;
   - Linux: -
   - Web: Cookies
3. For each identifier, ask the controller if the identifier may be used by an entity external to the messenger service;
4. Verify that each identifier is not used by an entity external to the messenger service;
5. Repeat the previous steps for all possible frontends;
6. If one or more identifiers are not specific to the application and the auditor considers they should be, ask the controller why;
7. If one or more identifiers may be used by an entity external to the messenger service and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all identifiers used by the application are specific to the application AND no identifier used by the application may be used by an entity external to the messenger service, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

IDENTIFIER_4 `basic`

The messenger application SHOULD use identifiers that are reset upon uninstalling the application.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all identifiers used by the application;
2. Verify that each identifier of the list is reset upon uninstalling the application. Such identifiers include, but are not limited to:
   - Android: Firebase installation ID (FID), globally-unique ID (GUID), App set ID, Advertising ID;
   - iOS: Vendor ID, Advertising ID;

- Windows: Advertising ID;
- Linux: -
- Web: Cookies

3. Repeat the previous steps for all possible frontends;
4. If one or more identifier is not reset upon uninstalling the application and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all identifiers used by the application are reset upon uninstalling the application, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

IDENTIFIER_5 `basic`

The messenger application SHOULD offer the user an option to reset the used identifiers.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all identifiers used by the application;
2. Mark down if the application features an option to reset the used identifiers;
3. Repeat the previous steps for all possible frontends;
4. If the application does not feature an option to reset the used identifiers and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the messenger features an option to reset the used identifiers, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.2 Privacy settings

SETTINGS_1 `basic`

Parameters associated with management of personal data SHOULD be grouped in a dedicated section or menu.

Prerequisite

- A populated account

Verification steps

1. Login with the account A;
2. Explore all the elements of the application interface and identify all parameters associated with management of personal data;
3. Verify that all parameters associated with the management of personal data are grouped in a dedicated section;
4. Verify that the section is clearly identified as dedicated to the management of personal data;
5. Repeat the previous steps for all possible frontends;
6. If not all the parameters associated with management of personal data are grouped in a dedicated section, and the auditor considers they should be, ask the controller why;
7. If the section is not clearly designated, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all parameters associated with management of personal data are grouped in a dedicated section AND the section is clearly designated, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SETTINGS_2 `basic`

The controller MAY choose to offer the user the option to configure the privacy settings and preferences during the account creation process.

Prerequisite

- None;

Verification steps

1. Create an account;
2. Mark down if during the account creation process the user is offered the option to configure the privacy settings;
3. Repeat the previous steps for all possible frontends;
4. If during the account creation process the user is not offered the option to configure the privacy settings, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, during the account creation process the user is offered the option to configure the privacy settings, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer to configure the privacy settings during the account creation process, the following requirements are not applicable: SETTINGS_2.a

SETTINGS_2.a `basic`

Privacy settings that are configurable during the account creation process MUST feature the most privacy friendly preset.

Prerequisite

- None;

Verification steps

1. Create an account;
2. Examine all the elements of the privacy setting section that are configurable during the account creation process;
3. Mark down if each setting is set to the most privacy-preserving state;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not offer to configure the privacy settings during the account creation process the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, all privacy settings configurable during the account creation process feature the most privacy-preserving preset. Otherwise, the requirement is not fulfilled (FAIL).

SETTINGS_3 `basic`

The privacy settings SHOULD be accessible anytime after the account creation/registration.

Prerequisite

- A populated account for each available frontend;

Verification steps

1. Proceed to the standard use procedure A.1.5;
2. Verify that during the standard use procedure, the privacy settings are accessible at anytime;
3. Repeat the previous steps for all possible frontends;
4. If at one or more time, the privacy setting is not accessible and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the privacy settings are accessible at anytime, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SETTINGS_4 `basic`

All privacy settings MUST be set to most privacy friendly setting by default.

Prerequisite

- None;

Verification steps

1. Create an account;
2. Examine all the elements of the privacy setting section after account creation;
3. Mark down if each setting is set to the most privacy-preserving state;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all privacy settings are set to the most privacy-preserving preset. Otherwise, the requirement is not fulfilled (FAIL).

2.9.3 Online status

ONST_1 `basic`

The visibility of the online status SHOULD be configurable by the user.

Preliminary steps

- A populated account A
- An account B that is contact of A

Verification steps

1. Login with the account A;
2. Verify that the visibility of the online status is configurable by the user;
3. Set the visibility of the online status to one of the available visibility options;
4. Login with the account B on another device;
5. Mark down if the visibility of the account A matches the setting set in step 3;
6. Repeat the previous steps for all available visibility settings;
7. Repeat the previous steps for all possible frontends;
8. If the visibility of the online status is not configurable by the user, and the auditor considers it should be, ask the controller why;

Validation

If the messenger does not feature an online status, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the visibility of the online status is configurable by the user AND in all cases the visibility is correctly applied AND includes a setting that renders the users visibility status to invisible to other users, OR if the auditor is satisfied with the

requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the visibility of the online status is not configurable by the user AND ONST_1 is fulfilled, the following requirements are not applicable (NA): ONST_1.a.

ONST_1.a `basic`

The online status MUST initially be set to be only visible to the user and nobody else.

Prerequisite

- None;

Verification steps

1. Create an account A;
2. Login to account A;
3. Verify that the online status is set to be only visible to the user and nobody else;
4. Repeat the previous steps for all possible frontends;

Validation

If the visibility of the online status is not configurable by the user AND ONST_1 is fulfilled, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the online status is set to be only visible to the user and nobody else. Otherwise, the requirement is not fulfilled (FAIL).

2.9.4 Read status of messages

READ_1 `basic`

The messenger SHOULD offer a way for a user to configure the visibility of the read status of messages the user received.

Preliminary steps

- A populated account A;
- An account B that is a contact of A;

Verification steps

1. Login with the account A;
2. Search for the option to configure the visibility of the read status of messages the user received;
3. Mark down if the option to to configure the visibility of the read status of messages the user received exists;
4. Disable the visibility of the read status of messages the user received;

5. Using the account B, send a message to A;
6. Using the account A, read the message sent by B;
7. Using the account B inspect the read status of the message sent to A;
8. Repeat the previous steps for all possible frontends;
9. If the option to configure the visibility of the read status of messages the user received does not exist, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature a read status indicator, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to configure the visibility of the read status of messages the user received exists AND user B could not determine if the message was read by A or not, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer to configure the visibility of the read status of messages the user received, the following requirements are not applicable: READ_1.a

READ_1.a `basic`

The visibility of read status of a message MUST initially be disabled.

Preliminary steps

- None;

Verification steps

1. Create an account A and login with A;
2. Inspect the setting panel where the visibility of read status of messages can be configured;
3. Mark down if the visibility of read status of messages is disabled;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature a read status indicator, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the visibility of read status of messages is disabled. Otherwise, the requirement is not fulfilled (FAIL).

READ_2 `basic`

The messenger SHOULD offer a way to modify the read status of a message the user received.

Preliminary steps

- A populated account A;

- An account B that is a contact of A;

Verification steps

1. Login with A;
2. Enable the visibility of the read status of all messages received by the user;
3. Using the account B, send a message to A;
4. Using the A, read the message sent by B;
5. Using A, search for the option to modify the read status of the message sent by B;
6. Mark down if the option to modify the read status of the message exists;
7. Using the account A, change the read status of the message to `not read`;
8. Using the account B, inspect the read status of the message sent to A;
9. Mark down if the read status of the message is `not read`;
10. Repeat the previous steps for all possible frontends;
11. If the option to modify the read status of the message does not exist, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature a read status indicator, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to modify the read status of the message exists AND the read status of the message is `not read`, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

READ_3 `basic`

The signaling regarding the read status of messages SHOULD be processed and handled in the frontend of the recipient.

Preliminary steps

- None;

Verification steps

1. Ask the controller to provide information on how the signaling of the read status is performed;
2. Verify that the signaling of the read status is processed and handled on the frontend of the recipient;
3. Ask the controller if, when the visibility of the read status is disabled, any information regarding the read status of a message leave the device at anytime;
4. Mark down if any information regarding the read status of a message leaves the device at anytime;

5. Repeat the previous steps for all possible frontends;
6. If the signaling of the read status is not processed and handled on the frontend of the recipient, and the auditor considers it should be, ask the controller why;
7. If information regarding the read status of a message leaves the device, and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature a read status indicator, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the signaling of the read status is processed and handled on the frontend of the recipient AND no information regarding the read status of a message leaves the device, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.5 Typing indicator

TYPE_1 `basic`

The messenger SHOULD offer a way for a user to configure the visibility of the typing indicator to other users.

Preliminary steps

- A populated account A connected on a device d_1;
- An account B that is a contact of A connected on a device d_2.

Verification steps

1. With the account A, search for the option to configure the visibility of the typing indicator to other users;
2. Mark down if the option to configure the visibility of the typing indicator to other users exists;
3. With the account A, disable the visibility of the typing indicator to other users;
4. With the account B, open the discussion panel with A and observe its content while performing the following step;
5. With the account A, start typing a message to B without sending it;
6. Mark down if any indication that A was typing a message appeared on the device used by B;
7. Repeat the previous steps for all possible frontends;
8. If the option to configure the visibility of the typing indicator to other users does not exist, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature a typing indicator the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to configure the visibility of the typing indicator to other users exists AND no indication that A was typing a message appeared,

OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer to configure the visibility of the typing indicator and this requirement is fulfilled (PASS), the following requirements are not applicable: TYPE_1.a

TYPE_1.a `basic`

The visibility of typing indicator MUST initially be disabled.

Preliminary steps

- None;

Verification steps

1. Create an account A;
2. Inspect the option to configure the visibility of the typing indicator;
3. Mark down if visibility of the typing indicator is disabled;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature a typing indicator, OR the typing indicator is not configurable AND requirement TYPE_1 is fulfilled (PASS) the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the visibility of the typing indicator is disabled. Otherwise, the requirement is not fulfilled (FAIL).

TYPE_2 `basic`

The signaling regarding the typing indicator SHOULD be processed and handled in the frontend of the sender.

Preliminary steps

- None;

Verification steps

1. Ask the controller to provide information on how the signaling of the typing indicator is performed;
2. Mark down that the signaling of the typing indicator is processed and handled on the frontend of the recipient;
3. Ask the controller if, when the visibility of the typing indicator is disabled, any information regarding the typing indicator leaves the device at anytime;
4. Mark down if any information regarding the typing indicator of a message leaves the device at anytime;
5. Repeat the previous steps for all possible frontends;

6. If the signaling of the typing indicator is not processed and handled on the frontend of the recipient, and the auditor considers it should be, ask the controller why;

7. If information regarding the typing indicator of a message leaves the device, and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature a typing indicator the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the signaling of the typing indicator is processed and handled in the frontend of the sender AND no information regarding the typing indicator of a message leaves the device, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.6 Profile data

PROF_1 `basic`

The visibility of the profile SHOULD be configurable by the user.

Prerequisite

- A populated account A connected on a device d_1;
- An account B that is a contact of A, connected on a device d_2.

Verification steps

1. With the account A, search for the option to configure the visibility of the profile to other users;
2. Mark down if the option to configure the visibility of the profile to other users exists;
3. With the account A, disable the visibility of the profile to other users;
4. With the account B, try to view the profile of user A;
5. Mark down if the profile of the account A was visible from the account B;
6. Repeat the previous steps for all possible frontends;
7. If the option to configure the visibility of the profile to other users does not exist, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature a profile the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to configure the visibility of the profile to other users exists AND of the account A was not visible from the account B, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer to configure the visibility of the profile and this requirment is fulfilled (PASS), the following requirements are not applicable (NA): PROF_1.a, PROF_1.b

PROF_1.a `basic`

The messenger MAY offer to configure the visibility of each profile related information such as profile description, picture etc. individually.

Preliminary steps

- A populated account A connected on a device d_1;

Verification steps

1. With the account A, search for the option to configure individually the visibility of each profile related information;
2. Mark down if, for each profile related information, its visibility can be configured individually;
3. Repeat the previous steps for all possible frontends;

4. If for one or more profile related information, the visibility cannot be configured individually, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature a profile OR does not offer to configure the visibility of the profile AND PROF_1 is fulfilled, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the visibility of indi all profile related information can be configured individually, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PROF_1.b `basic`

If the messenger offers a way to configure visibility settings individually, there MUST be a setting to set the visibility for all profile related information in a single setting.

Preliminary steps

- A populated account A connected on a device d_1;

Verification steps

1. With the account A, search for the option to configure the visibility for all profile related information in a single setting;
2. Mark down if the option to configure the visibility for all profile related information in a single setting exists;
3. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature a profile OR the messenger does not offer a way to configure visibility settings individually, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to configure the visibility for all profile related information in a single setting exists. Otherwise, the requirement is not fulfilled (FAIL).

PROF_2 `basic`

The profile MUST initially be set to be only visible to the user and nobody else.

Preliminary steps

- None;

Verification steps

1. Create an account A and login with A;
2. Inspect the setting panel where the visibility of the profile can be configured;
3. Mark down if the visibility of the profile is set so that it is only visible to the user and nobody else;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature a profile, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the visibility of the profile is set so that it is only visible to the user and nobody else. Otherwise, the requirement is not fulfilled (FAIL).

PROF_3 `basic`

The messenger SHOULD feature an option to set the elements of the profile to a default value.

Preliminary steps

- An account A connected on a device d_1;

Verification steps

1. With the account A, fill in all the elements of the profile with a custom value;
2. For each element of the profile, search for the option to set it to a default value;
3. Mark down if, for each element of the profile, the option to set it to a default value exists;
4. For each element of the profile, use to option to set it to a default value;
5. For each element of the profile, inspect the current value;
6. Mark down if, for all element of the profile, the value has been reverted to the default one;
7. Repeat the previous steps for all possible frontends;
8. If for one or more elements of the profile, the option to set it to a default value does not exist, and the auditor considers they should, ask the controller why;

Validation

If the messenger does not feature a profile, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, there exists a option to set every individual element of the profile to a default value AND all elements of the profile the value has been reverted to the default one, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.7 Link preview

PREVIEW_1 `basic`

The messenger SHOULD feature a way to enable or disable the link preview feature.

Preliminary steps

- A populated account A connected on a device d_1;
- An account B that is a contact of A connected on a device d_2;

Verification steps

1. With the account A, search for the option to disable the link preview feature;
2. Mark down if the option to disable the link preview feature exists;
3. With the account A, disable the link preview feature;
4. With the account B, send a link in a message to A;
5. With the account A, inspect the discussion with B;
6. Mark down if the preview of the link is displayed on the device of A;
7. Repeat the previous steps for all possible frontends;
8. If the option to disable the link preview feature does not exist, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature link preview, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to disable the link preview feature exists AND the preview of the link is not displayed, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PREVIEW_2 `basic`

The link preview feature SHOULD initially be disabled.

Preliminary steps

- None;

Verification steps

1. Create an account A and login with A;
2. Inspect the setting panel where link preview feature can be configured;
3. Mark down if the link preview feature is disabled;
4. Repeat the previous steps for all possible frontends;
5. If the link preview feature is not disabled, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature link preview, OR link preview can not be configured AND PREVIEW_2 is fulfilled (PASS) the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the link preview feature is disabled, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.8 Multimedia content metadata

METADATA_1 `basic`

The messenger SHOULD offer the feature to strip the metadata from multimedia content before sending them.

Prerequisite

- A populated account , A;
- A picture p with metadata;
- A software able to inspect metadata[a];

Verification steps

1. Login with account A;
2. Search for the feature to strip the metadata from multimedia content;
3. Verify that the messenger offers a feature to strip the metadata from multimedia content;
4. Enable the feature;
5. Select B, a contact of A;
6. Send the picture to B;
7. Login with account B;
8. Save the picture on the device;
9. Inspect the metadata of the picture using the appropriate software;
10. Verify that the picture does not contain metadata. The auditor should especially verify that the following categories of metadata are not present:
    - GPS coordinates;
    - Date and time the picture was taken;
    - Device model, type, and manufacturer;
    - Name and version of editing tools used to modify the picture;
11. Repeat the previous steps for all possible frontends;
12. If the messenger does not offer the feature to strip the metadata from the picture, and the auditor considers it should, ask the controller why;
13. If one or more metadata are contained in the picture, and the auditor considers they should not be, ask the controller why;

Validation

If the messenger does not support any type of multimedia content the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the messenger offers the feature to strip the metadata from the picture AND no metadata is contained in the picture, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer the feature to strip the metadata from multimedia content AND this requirement is fulfilled, the following requirements are not applicable (NA): METADATA_2 and METADATA_3.

[a]For instance Gimp: https://www.gimp.org/ (`Image > Metadata > View Metadata`)

METADATA_2 basic

The feature to strip metadata from multimedia content before sending SHOULD be enabled by default.

Prerequisite

- None;

Verification steps

1. Create an account;
2. Verify that the feature to strip metadata from multimedia content before sending is enabled;
3. Repeat the previous steps for all possible frontends;
4. If the feature to strip metadata from multimedia content before sending is not enabled, and the auditor considers it should be, ask the controller why;.

Validation

If the messenger does not offer the feature to strip the metadata from multimedia content AND METADATA_1 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the feature to strip metadata from multimedia content before sending is enabled, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

METADATA_3 basic

The messenger SHOULD inform the user before stripping metadata from multimedia content.

Prerequisite

- A populated account , A;
- a picture with metadata;

Verification steps

1. Login with the account A;
2. Ensure that the feature to strip metadata from multimedia content before sending is enabled;
3. Send the picture to one of the contacts of A;
4. Verify that, before sending the picture, the user is informed that the metadata is stripped;
5. Repeat the previous steps for all possible frontends;

6. If the user is not informed that the metadata is stripped, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not offer the feature to strip the metadata from multimedia content AND METADATA_1 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user is informed that the metadata is stripped before it is striped, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.9 Communication with other applications

SHARE_1 `basic`

Prior to sharing personal data with other applications running on the device the messenger application MUST acquire consent.

Prerequisite

- A populated account ;
- A device for each available frontend;;

Verification steps

1. Ask the controller to provide a list of all situations where the messenger application might share personal data with other applications running on the device;
2. For each listed situation perform the following:
   (a) Login with the account;
   (b) Use the application to reach one listed situation that was not tested in a previous iteration;
   (c) Verify that, before reaching the described situation, the consent of the user is requested;
3. Ask the controller, for each context in which the messenger application might share personal data with other applications running on the device, what happens if the user does not provide consent;
4. Verify that, for each context in which the messenger application might share personal data with other applications running on the device, if the consent of the user is not obtained, no personal data is shared;
5. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all context where a personal data can occur, the consent of the user is requested before reaching the context AND no personal data is transferred if the consent is not obtained. Otherwise, the requirement is not fulfilled (FAIL).

2.9.10 Access control to device resources

PERM_1 `basic`

The messenger application MUST only request individual permissions to access resources from the device that are necessary for the delivery of its declared functionalities.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of permission requested by the application as well as a list of declared functionalities;
2. Ask the controller to provide, for each permission, the associated functionalities as well as a justification for this association;
3. Verify, for each permission, that it is necessary for the delivery of at least one declared functionality;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all permission requested by the application are necessary for the delivery of at least one declared functionality, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PERM_2 `basic`

The description of the messenger application SHOULD declare the permissions that the application might request at runtime.

Prerequisite

- The description of the messenger application for each available frontend;

Verification steps

1. Ask the controller to provide the list of the permissions that the application might request at runtime;
2. Inspect the description of the application and search for the list of permissions that the application might request at runtime;
3. Verify that each permission that the application might request at runtime is declared in the description;
4. Repeat the previous steps for all possible frontends;
5. If one ore more permissions are not declared in the description, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all the permission that the application might request at runtime are declared in the description, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PERM_3 `basic`

For each requested permission, the messenger application MUST inform the user of the necessity of the permission requested.

Prerequisite

- The description of the messenger application for each available frontend;

Verification steps

1. Ask the controller to provide the list of permission requested by the application;
2. Inspect the description and search for the information about the necessity of the permission requested by the application;
3. Inspect the permission prompt of each requested permission by the application and search for the information about the necessity of the permission requested by the application
4. Mark down, for each requested permission, if its necessity is clearly and truthfully explained;
5. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all permission requested by the application their necessity is clearly and truthfully explained, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PERM_4 `basic`

The messenger application MUST NOT circumvent permissions restrictions. (Ex.: using Wi-Fi/Bluetooth permission to infer geolocation).

Prerequisite

- None;

Verification steps

1. Ask the controller if the application circumvent permission restrictions. Circumventing permission restrictions include, but is not limited to :
   - Using a permission to infer information that is protected by another permission;
   - Obtaining information usually protected by a permission, but without requesting this permission;

2. Mark down if the controller declared that the application does not circumvent permission restrictions;

3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller declared that the application does not circumvent permission restrictions. Otherwise, the requirement is not fulfilled (FAIL).

PERM_5 `basic`

The messenger SHOULD request access to a resource only when a feature requested by the user requires the access to the resource.

Prerequisite

- A device for each available frontend;

Verification steps

1. Ask the controller to provide a list of the resources that can be requested by the application along with the declared feature that uses it;
2. Create a new account A;
3. For each declared feature perform the following:
   (a) Login with the account;
   (b) Use the application to reach one declared feature that was not tested in a previous iteration;
   (c) Verify that the request by the application corresponding to the declared feature is only performed immediately prior to using the declared feature for the first time;
4. Repeat the previous steps for all possible frontends;
5. If for one or more resources, the access is requested prior to when the feature is requested by the user, and the auditor considers it should not be, ask the controller why;
6. If for one or more resource, the access to the resource is not required for a feature requested by the user, and the auditor considers it should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all resources the access is requested not earlier than it is needed AND is required by a feature requested by the user, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PERM_6 `basic`

The messenger SHOULD request only one permission at a time.

Prerequisite

- A populated account
- A device for each available frontend;

Verification steps

1. Ask the controller to provide a list of the resources that can be requested by the application along with the declared feature that uses it;
2. For each declared feature perform the following:
   (a) Login with the account;
   (b) Use the application to reach one declared feature that was not tested in a previous iteration;
   (c) Mark down if the request by the application corresponding to the declared feature is only requesting access to a single resource at a time;
3. Repeat the previous steps for all possible frontends;
4. If the application requests more than one permission at a time, and the auditor considers it should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application did not request more than one permission at a time, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PERM_7 `basic`

When requesting access to a resource, the messenger MAY offer a way to remember the choice of the user and apply it to all future requests.

Prerequisite

- A populated account
- A device for each available frontend;

Verification steps

1. Ask the controller to provide a list of the resources that can be requested by the application along with the declared feature that uses it;
2. For each declared feature perform the following:
   (a) Login with the account;
   (b) Use the application to reach one declared feature that was not tested in a previous iteration;
   (c) Mark down if the request by the application corresponding to the declared feature offers a way to remember the choice for future requests;
   (d) Select the option not to remember it and grant the request;
   (e) Use the application to reach the same declared feature;
   (f) Mark down if the same request appears again;

(g) Select the option to remember it and deny the request;

(h) Use the application to reach the same declared feature;

(i) Mark down if the same request appears again;

3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all resources the application does offer a way to remember the choice of the user and apply it to all future requests AND the application does request access again in the second iteration while don't requesting access again in the third for all declared features, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PERM_8 `basic`

In case the user refuses to grant access to a resource, the messenger SHOULD continue to work, potentially in a degraded mode.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of the resources that can be requested by the application along with the declared feature that uses it;
2. Ask the controller, for each resource, what happens if the user refuses to grant access to the resource;
3. Mark down if, for each resource, the application continues to work if the user refuses to grant access;
4. Repeat the previous steps for all possible frontends;
5. If for one or more resources the application stops working in case the user refuses to grant access, and the auditor considers it should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all resources the application continues to work in an acceptable way if the user refuses access, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.11 Application behaviour

APP_BEHAV_1 `basic`

The messenger MAY feature a mechanism to blank the display of the current application state while switching between applications.

Prerequisite

- A device for each available frontend;;
- An account A connected on each device;

Verification steps

1. Mark down if the application offers a feature to protect its content while switching applications;
2. Enable the feature;
3. Put the application in foreground;
4. Put the application in background;
5. Observe the preview of the application by using the application switcher or any similar mechanisms;
6. Mark down if the preview of the application reveals any content of the current state of the application;
7. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the preview of the application does not reveal any content of the current state of the application. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer the feature to blank the display of the current application state while switching between applications, the following requirements are not applicable (NA): ABB_BEHAV_1.a.

APP_BEHAV_1.a `basic`

The feature to protect the application screen while switching between recent applications SHOULD be enabled by default.

Prerequisite

- None;

Verification steps

1. Create a new account A;
2. Mark down the default value of the option to protect the current state of the application while switching applications;

3. Repeat the previous steps for all possible frontends;
4. If the option is not enabled by default, and the auditor considers it should be, ask the controller why;

Validation

If the application does not feature an option to protect the current state of the application while switching application, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to protect the current state of the application while switching applications is enabled by default, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

APP_BEHAV_2 `basic`

The messenger MAY feature a protection against screenshot capture.

Prerequisite

- A device for each available frontend;;
- an account *A* connected on each device;

Verification steps

1. Put the application in foreground;
2. Try to take a screenshot;
3. Mark down if the screenshot failed;
4. Repeat the previous steps for all possible frontends;
5. If an actual screenshot of the application interface was taken, and the auditor considers it should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the screenshot failed, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

APP_BEHAV_3 `basic`

The messenger SHOULD offer the option to lock the application with an individual code, passphrase or device feature.

Prerequisite

- A device for each available frontend;;
- an account *A* connected on each device;

Verification steps

1. Search for the option to lock the application with an individual code, passphrase or device feature;
2. Mark down if the option exits;
3. Enable the option to lock the application by using on of the available options;
4. Put the application in background;
5. Put the application in foreground;
6. Mark down if the application is locked;
7. Repeat the previous steps for all available options to lock the device;
8. Repeat the previous steps for all possible frontends;
9. If the option does not exist, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the option to lock the application exists AND the application is locked, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer the option to lock the application with an individual code, passphrase or device feature, the following requirements are not applicable (NA): APP_BEHAV_3.a.

APP_BEHAV_3.b `basic`

The messenger MUST NOT solely use device features in order to provide the feature.

Prerequisite

- For each available frontend;
  - an account A connected on a device d_1;

Verification steps

1. Search for the option to lock the application with an individual code, passphrase or device feature;
2. Mark down all available options;
3. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not offer the option to lock the application with an individual code, passphrase or device feature, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the device locking feature can be used without having to use a device feature. Otherwise, the requirement is not fulfilled (FAIL).

APP_BEHAV_4 `basic`

The messenger MUST offer the option to lock the application with an individual code, passphrase or device feature if the messenger is primarily intended to be used with Art. 9 GDPR Data.

Prerequisite

- A device for each available frontend;
- an account A connected on each device;

Verification steps

1. Search for the option to to lock the application with an individual code, passphrase or device feature;
2. Mark down if the option exists;
3. Repeat the previous steps for all possible frontends;

Validation

If the messenger is not primarily intended to be used with Art. 9 GDPR Data, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to lock the application exists. Otherwise, the requirement is not fulfilled (FAIL).

APP_BEHAV_5 `basic`

The messenger SHOULD NOT display personal data if the user has not requested it.

Prerequisite

- None;

Verification steps

1. Ask the controller for a list of prompts that might be shown in the application without direct user interaction;
2. Mark down if any of these prompts include personal data. This includes, but is not limited to, an autonomous pop-up request to validate a contact e-mail or phone number that directly displays the identifier;
3. Repeat the previous steps for all possible frontends;
4. If the application can display personal data without the user having requested it, and the auditor considers it should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, non of the automated prompts contain any personal data, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

APP_BEHAV_6 basic

While running in the background, the messenger application SHOULD NOT collect data other than those strictly necessary to the messenger functionalities.

Prerequisite

- None;

Verification steps

1. Ask the controller the list of data that is collected by the application while running in background;
2. Ask the controller, for each data, to justify the strict necessity for the messenger functionalities of the background collection;
3. Repeat the previous steps for all possible frontends;
4. If one or more data not strictly necessary to the messenger functionalities are collected in background, and the auditor considers they should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all data collected in background is strictly necessary to the messenger functionalities, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.12 Identity manager

ID_MGR_1 basic

The messenger SHOULD offer account creation independent of third party identity managers.

Prerequisite

- None;

Verification steps

1. Start the account creation process;
2. Mark down if there is an option to create the account without using or interacting with a third party identity managers;
3. Create an account without using or interacting with any third party identity manager;
4. Verify that the account was successfully created;
5. Repeat the previous steps for all possible frontends;
6. If there is no option to create the account without using or interacting with a third party identity managers, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, there is an option to create the account without using or interacting with a third party identity managers AND the account was successfully created, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ID_MGR_1.a `basic`

The messenger SHOULD offer a login process independent of third party identity managers.

Prerequisite

- An account A.

Verification steps

1. Start the login process with account A;
2. Mark down if there is an option to login without using or interacting with a third party identity manager;
3. Login with the account A without using or interacting with a third party identity manager;
4. Mark down if the user is logged in;
5. Repeat the previous steps for all possible frontends;
6. If there is no option to login without using or interacting with a third party identity manager, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, there is an option to login the account without using or interacting with a third party identity manager AND the user is logged in, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ID_MGR_2 `basic`

The messenger SHOULD offer a way to anonymously create an account.

Prerequisite

- None;

Verification steps

1. Create an account on the messenger service without providing any personally identifiable information. Personally identifiable information includes, but in not limited to: name, E-mail address, phone number...
2. Verify that the account was successfully created;
3. Repeat the previous steps for all possible frontends;

4. If there is no way to create an account without providing any personal information and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the messenger provides a away to create an account anonymously AND the account was successfully created, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.13 Push notifications

PUSH_1 `basic`

Push notifications MUST be configurable by the user.

Prerequisite

- A populated account

Verification steps

1. Search for the option to disable or enable push notifications in the settings panel;
2. Mark down if the option to disable or enable push notifications exists;
3. Create an account B, and add A as contact;
4. With account B disable push notifications;
5. On the device used by account B, put the application in the background;
6. With account A, send a message to B;
7. On the device used by account B, inspect the notifications;
8. Mark down if there is a notification associated to the reception of the message;
9. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature push notifications the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to disable or enable push notifications exists AND B has not received a notification associated to the reception of the message. Otherwise, the requirement is not fulfilled (FAIL).

PUSH_1.a `basic`

The push notification MUST initially be disabled.

Prerequisite

- None;

Verification steps

1. Create an account A;
2. Mark down the default value of the option to configure push notifications;
3. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature push notifications the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to configure push notifications is set to disabled. Otherwise, the requirement is not fulfilled (FAIL).

PUSH_2 `basic`

The contents of the push notification displayed SHOULD be configurable by the sending user.

Prerequisite

- A populated account A;
- An account B that is contact of A;

Verification steps

1. Search for the option to configure the content of push notifications of outgoing messages in the settings panel. Elements of the notification that should be configurable through this option includes, but are not limited to: content the message and identity of the sender;
2. Mark down if the option to configure the content of push notifications of outgoing messages exists;
3. On the device used by account B, enable push notification, and put the application in background;
4. On the device used by account A, configure push notification of outgoing messages such that the content of the message and the identity of the sender are not included;
5. With account A, send a message to B;
6. On the device used by account B, inspect the notifications;
7. Mark down if the content of the message is included in the notification;
8. Mark down if the identity of the sender is included in the notification;
9. Repeat the previous steps for all possible frontends;
10. If the option to configure the content push notifications of outgoing messages does not exist, and the auditor considers it should, ask the controller why;
11. If the content of the message is included in the notification, and the auditor considers it should not be, ask the controller why;
12. If the identity of the sender is included in the notification, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature push notifications the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to configure the content of push notifications of outgoing messages exists AND the content of the message is not included in the notification AND the identity of the sender is not included in the notification, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PUSH_3 `basic`

The contents of the push notification SHOULD be configurable by the receiving user.

Prerequisite

- A populated account A;
- An account B that is contact of A;

Verification steps

1. Search for the option to configure the content of push notifications of incoming messages in the settings panel. Elements of the notification that should be configurable through this option includes, but are not limited to: content the message and identity of the sender;
2. Mark down if the option to configure the content push notifications of incoming messages exists;
3. On the device used by account A, configure push notification of incoming messages such that the content of the message and the identity of the sender are not included;
4. On the device used by account A, enable push notification, and put the application in background;
5. With account B, send a message to A;
6. On the device used by account A, inspect the notifications;
7. Mark down if the content of the message is included in the notification;
8. Mark down if the identity of the sender is included in the notification;
9. Repeat the previous steps for all possible frontends;
10. If the option to configure the content push notifications of incoming messages does not exist, and the auditor considers it should, ask the controller why;
11. If the content of the message is included in the notification, and the auditor considers it should not be, ask the controller why;
12. If the identity of the sender is included in the notification, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature push notifications the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to configure the content

of push notifications of incoming messages exists AND the content of the message is not included in the notification AND the identity of the sender is not included in the notification, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

PUSH_4 `basic`

The controller MUST prove that the used push notification service complies with the GDPR.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of push notification services used by the application;
2. Ask the controller to provide, for each push notification service, evidence proving the compliance of the service provider with the GDPR.
3. Verify if the evidence provided by the controller to prove the compliance of the service provider with the GDPR. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature push notifications the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the evidence provided by the controller prove the compliance of the service provider with the GDPR. Otherwise, the requirement is not fulfilled (FAIL).

2.9.14 Contact matching

CONTACT_1 `basic`

The controller MUST provide a valid legal basis for processing contact data obtained from the user.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a legal basis for processing contact data obtained from the user;
2. Verify that the controller has provided a valid legal basis for processing contact data obtained from the user.
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller has provided a valid legal basis for processing contact data obtained from the user. Otherwise, the requirement is not fulfilled (FAIL).

CONTACT_2 `basic`

The messenger application SHOULD NOT send a complete copy of the contacts from the contact book to the backend server.

Prerequisite

- None;

Verification steps

1. Ask the controller if it accesses the contact book of the device;
2. Ask the controller if the any data collected from the contact book of the device is sent to the backend server;
3. Repeat the previous steps for all possible frontends;
4. If one or more data collected from the contact book is sent to the backend server, and the auditor considers they should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no data from the contact book of the device is sent to the backend server, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CONTACT_3 `basic`

The contact identifiers sent to the backend server SHOULD be encoded in a way that feature brute-force protection.

Prerequisite

- None;

Verification steps

1. Ask the controller how the contact identifiers are encoded before being sent to the backend server in the context of contact matching.
2. Verify that the encoding scheme used provides a level of protection against bruteforce enumeration. Such protection can be achieved by using *key stretching* techniques or other techniques that rely on encoding functions that are slow to execute. Such functions include, but are not limited to, Password-Based Key Derivation functions [ENI14a, Sec. 5.1] .
3. Repeat the previous steps for all possible frontends;
4. If the encoding scheme does not provide a level of protection against bruteforce enumeration, and the auditor considers it should, ask the controller why;.

Validation

If the messenger does not feature contact matching the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the encoding scheme provides a level of protection against bruteforce enumeration, OR if the auditor is satisfied with the requested

explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CONTACT_4 `basic`

The messenger SHOULD offer the option to store individual contacts in the messenger without forcing the usage of the device contact list.

Prerequisite

- Two accounts, A & B that are not mutual contacts.

Verification steps

1. With the account A, search for an option to store contacts locally without using the device contact list;
2. Mark down if this option exists;
3. Set the option to store new contacts in a local storage without using the device contact list;
4. With the account A, add B as a contact;
5. On the device used by the account A, inspect the contact list of the device and search for the account B;
6. Mark down if any data of the account B is in the contact list of the device;
7. Repeat the previous steps for all possible frontends;
8. If the messenger does not feature an option to store contacts in a local storage without using the device contacts list and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the messenger features an option to store contacts in a local storage without using the device contact list AND the account B is not in the device contact list, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.15 Groups

GROUPS_1 `basic`

A user's personal information SHOULD NOT be visible to other users in this group who do not already know the users personal information.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - an account B that is not a contact of A, connected on a device d_2;

Verification steps

1. Select one group G to which A belongs;
2. With the account B, join the group G;
3. With the account B, inspect the user A and mark down the available information on the account A;
4. With the account A, inspect the user B and mark down the available information on the account$B$;
5. Repeat the previous steps for all possible frontends;
6. If one or more personal data of account A are visible from account B, and the auditor considers they should not be, ask the controller why;
7. If one or more personal data of account B are visible from account A, and the auditor considers they should not be, ask the controller why;

Validation

If the messenger does not feature groups, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, no personal data (other than the username) of the account A is visible from the account B AND no personal data (other than the username) of the account B is visible from the account A OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_2 `basic`

The application SHOULD feature an option for the user to prevent other users from adding them to a group.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - an account B who is a contact of A, connected on a device d_2;

Verification steps

1. Using the account A, search for the setting that allows configuring if the user can be added to groups by another user;
2. Mark down if this setting exists;
3. Using the account A and this setting, disallow B to add A to a group;
4. Using the account B, create a new group G, and attempt to add A to the group G;
5. Mark down if A is added to G;
6. Repeat the previous steps for all possible frontends;

7. If the setting to configure if the user can be added to a group by another user does not exist and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature groups, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the setting to configure if the user can be added to a group by another user exists AND A could not be added to the group G, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the option to configure if the user can be added to groups by another user does not exist, the following requirements are not applicable (NA): GROUPS_2.a.

GROUPS_2.a `basic`

By default, the setting to configure if another user can add them to a group SHOULD be set in a way that this is disabled for all other users.

Prerequisite

- An account A connected on a device d_1;

Verification steps

1. Create a new account B and add A as a contact;
2. Using the account B, search for the setting to configure if the user can be added to groups by another user;
3. Mark down the current setting;
4. Using the account A, attempt to add B to a group G;
5. Mark down if B is added to G;
6. Repeat the previous steps for all possible frontends;
7. If the default setting to configure if the user can be added to groups by another user allows adding the user in any tested frontend and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature groups OR if the messenger does not feature the option to configure if a user can be added to groups by another user AND GROUPS_2 is fulfilled (PASS), the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the default setting to configure if the user can be added to groups by another user is set so that no other user is able to add the tested user to a group, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_3 `basic`

Prior to being added to a group the user SHOULD be asked for consent.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B who is a contact of A, connected on a device d_2;

Verification steps

1. With the account B, create a new group G and add A to the group;
2. Using the account A, inspect if it is in the group G;
3. Mark down if the application requests the consent of A to join the group;
4. Repeat the previous steps for all possible frontends;
5. If the account A is in the group, and the auditor considers it should not be, ask the controller why;
6. If the application does not request the consent of A to join the group, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature groups, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the account A is not in the group AND the application requests the consent of A to join the group, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_3.a `basic`

The messenger MAY enable the user to configure who can add them to a group without their consent.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B who is a contact of A, connected on a device d_2;

Verification steps

1. With the account A, search for the feature to enable another user to add them to a group without requesting their consent;
2. Mark down if this feature exists;
3. With the account A, enable this feature for B;
4. Using the account B, create a new group G and add A to the group;

5. Using the account A, inspect if it is in the group G;
6. Mark down if A is in the group G;
7. Repeat the previous steps for all possible frontends;
8. If A is not in the group, and the auditor considers it should be, ask the controller why;

Validation

If the messenger does not feature groups, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the feature to enable another user to add them to a group without requesting their consent exists AND A is in the group, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_4 `basic`

New members of a group SHOULD NOT be able to see previous members of a group.

Prerequisite

- For each available frontend:
  - Four accounts A, B, C, D that are not mutual contacts, each of them connected on a device.

Verification steps

1. With account A, create and join the group G;
2. Join the group G with account B and C;
3. For each account A, B, C, send one message to the group G;
4. Leave the group with account B and C;
5. Join the group G with account D;
6. Mark down if, from the account D, the information associated to the group includes any identifiers of accounts B or C;
7. Repeat the previous steps for all possible frontends;
8. If the information associated to the group includes one or more identifiers of accounts B or C, and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature groups, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the information associated to the group seen by D does not include any identifiers of accounts B and C, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_5 `basic`

The messenger SHOULD NOT offer a feature to display the list of recently left members of a group.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B that is not a contact of A and is not sharing a group with A, connected on a device d_2;

Verification steps

1. With the account B, join a group G to which A belongs;
2. Send a message to the group G with account B;
3. Leave the group G with account B;
4. With the account A search, within the information associated to the group G, for a feature to display the list of recently left members of the group;
5. Mark down if a feature to display the list of recently left members of the group exists;
6. Repeat the previous steps for all possible frontends;
7. If a feature to display the list of recently left members of the group exists and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature groups, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, a feature to display the list of recently left members of the group does not exist, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer a feature to display the list of recently left members of a group, the following requirements are not applicable (NA): GROUPS_5.a, GROUPS_5.b.

GROUPS_5.a `basic`

When leaving the group, users MUST be offered the choice not to appear in the list of recently left members of the group.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B that is sharing a group G with A, connected on a device d_2;

Verification steps

1. With the account A leave the group G;
2. Mark down if the choice not to appear in the list of recently left members of the G is offered to the user;
3. Select the option not to appear in the list of recently left members of the group;
4. With the account B, inspect the list of recently left members of the group G;
5. Mark down if A is in the list of recently left members of the group G;
6. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature groups OR if the messenger does not offer a feature to display the list of recently left members of a group the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option not to appear in the list of recently left members of the group is offered to the leaving user AND if A is not in the list of recently left members of the group. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_5.b `basic`

Users listed in the list of recently left members of a group SHOULD NOT be listed longer than 72h after leaving.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B that is sharing a group G with A, connected on a device d_2;

Verification steps

1. With the account A leave the group G;
2. If prompted, choose to appear in the list of recently left members;
3. Wait 72 hours;
4. With the account B, inspect the list of recently left members of the group G;
5. Mark down of A is in the list of recently left members of the group G;
6. Repeat the previous steps for all possible frontends;
7. If A is in the list of recently left members of the group, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature groups OR if the messenger does not offer a feature to display the list of recently left members of a group the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, A is not in the list of recently left members of the group, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_6 `basic`

The groups message history SHOULD NOT be visible to new members.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B that is not a contact of A and is not sharing a group with A, connected on a device d_2;;

Verification steps

1. With the account B, join G a group to which A belongs;
2. With the account B, inspect the message history of G;
3. Mark down if, from the point of view of account B, one or more messages are visible in the group G;
4. Repeat the previous steps for all possible frontends;
5. If one or more messages are visible and the auditor considers they should not be, ask the controller why;

Validation

If the messenger does not feature groups, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, no message is visible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_7 `basic`

The visibility of a groups message history to new members MAY be configurable by the group admin.

Prerequisite

- For each available frontend:
  - An account A that is admin of a group G, connected on a device d_1;
  - Two accounts B and C that are not in G, connected on a device d_2;

Verification steps

1. With the account A, search for the feature to configure the visibility of a groups message history to new members for group G;
2. Mark down if the feature to configure the visibility of a groups message history to new members exists;
3. With the account A disable the visibility of a groups message history to new members in group G;
4. With the account A, send a message m_1 to G;

5. With the account B, join the group G;
6. Mark down if, from the point of view of account B, one ore more messages are visible in the group G;
7. With the account A, enable the visibility of a groups message history to new members;
8. With the account A, send a message m_2 to G;
9. With the account C, join the group G;
10. Mark down if, from the point of view of account C, one ore more messages are visible in the group G;
11. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature groups OR does not feature group message history OR does not feature setting the visibility of group message history to new members OR the concept of group admin does not exist, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the feature to configure the visibility of a groups message history to new members exists AND from the point of view of account B no messages are visible AND from the point of view of account C only message m_2 and not m_1 is visible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the feature to configure the visibility of a groups message history to new members does not exist, the following requirements are not applicable (NA): GROUPS_7.a.

GROUPS_7.a `basic`

This configuration SHOULD initially be set so that the history is not visible to new members.

Prerequisite

- An account A that is admin of a group G, connected on a device d_1;

Verification steps

1. With the account A, create a group G;
2. With the account A search for the feature to configure the visibility of a groups message history to new members;
3. Mark down if the visibility of a groups message history to new members is disabled;
4. Repeat the previous steps for all possible frontends;
5. If the visibility of a groups message history to new members is not disabled, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature groups OR does not feature group message history, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the

visibility of a groups message history to new members is disabled, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_8 `basic`

The messenger MAY offer a feature to display a limited number of past group messages to new members of a group.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B that is not a contact of A and is not sharing a group with A, connected on a device d_2;

Verification steps

1. With the account A, select a group G to which A belongs and search for the setting that allows to set a number of past group messages to new members of the group;
2. Mark down the maximum number of past messages that can be set to be visible to new members of the group G;
3. With the account A, set the maximum number to be visible to new members of the group G;
4. With the account A send 101 messages to the group G;
5. With the account B, join the group G and inspect the messages of the group G;
6. Mark down the number of messages of the group G visible to the account B;
7. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature groups OR does not feature group message history OR does not feature setting the visibility of group message history to new members, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the number of messages of the group G visible to the account B is equal or less than 100, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_9 `basic`

Upon leaving a group, the past messages of the user MAY be masked as messages from an anonymous past member instead of appearing with that user's identifier.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a group G with A, connected on a device d_2;

Verification steps

1. With the account A, send a message on the group G
2. Leave the group G with account A;
3. Inspect the message history of the group G;
4. Mark down if for all messages authored by A in the group G their author is set to a default value;
5. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature groups OR does not feature group message history, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, all messages authored by A in the group G have their author set to a default value, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

GROUPS_10 `basic`

The messenger SHOULD offer the option to exit a group without notifying the users of the group.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a group G with A, connected on a device d_2;

Verification steps

1. With the account A, leave the group G;
2. Mark down if A was prompted to leave the group without notifying the other users of the group;
3. Select not to notify other members of the group;
4. With the account B, inspect potential notifications and information associated to the group G;
5. Mark down if B is notified that A has left the group G;
6. Repeat the previous steps for all possible frontends;
7. If A is not promped to leave the group silently, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature groups, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, A is prompted to leave the group G silently AND B is not notified that A has left the group G, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.16 Communities

COMMUNITIES_1 `basic`

A users personal information SHOULD NOT be visible to other users in a community who do not already know the users personal information.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - an account B that is not a contact of A, connected on a device d_2;

Verification steps

1. Select one community C to which A belongs;
2. With the account B, join the community C;
3. With the account B, inspect the user A and mark down the available information on the account A;
4. With the account A, inspect the user B and mark down the available information on the account B;
5. Repeat the previous steps for all possible frontends;
6. If one or more personal data of account A are visible from account B, and the auditor considers they should not be, ask the controller why;
7. If one or more personal data of account B are visible from account A, and the auditor considers they should not be, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, no personal data (other than the username) of the account A is visible from the account B AND no personal data (other than the username) of the account B is visible from the account A OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_2 `basic`

The application SHOULD feature an option for the user to prevent other users from adding them to a community.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - an account B who is a contact of A, connected on a device d_2;

Verification steps

1. Using the account A, search for the setting that allows configuring if the user can be added to communities by another user;
2. Mark down if this setting exists;
3. Using the account A and this setting, disallow B to add A to a community;
4. Using the account B, create a new community G, and attempt to add A to the community G;
5. Mark down if A is added to G;
6. Repeat the previous steps for all possible frontends;
7. If the setting to configure if the user can be added to a community by another user does not exist and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the setting to configure if the user can be added to a community by another user exists AND A could not be added to the community G, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the option to configure if the user can be added to communities by another user does not exist, the following requirements are not applicable (NA): COMMUNITIES_2.a.

COMMUNITIES_2.a `basic`

By default, the option to configure if another user can add them to a community SHOULD be disabled for all other users.

Prerequisite

- An account A connected on a device d_1;

Verification steps

1. Create a new account B and add A as a contact;
2. Using the account B, search for the setting to configure if the user can be added to communities by another user;
3. Mark down the current setting;
4. Using the account A, attempt to add B to a community CO;
5. Mark down if B is added to CO;
6. Repeat the previous steps for all possible frontends;
7. If the default setting to configure if the user can be added to communities by another user allows adding the user in any tested frontend; and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature communities OR if the messenger does not feature the option to configure if a user can be added to communities by another user, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the default setting to configure if the user can be added to communities by another user is set so that no other user is able to add the tested user to a community, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_3 `basic`

Prior to being added to a community the user SHOULD be asked for consent.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B who is a contact of A, connected on a device d_2;

Verification steps

1. With the account B, create a new community C and add A to the community;
2. Using the account A, inspect if it is in the community C;
3. Mark down if A is in the community;
4. Mark down if the application requests the consent of A to join the community;
5. Repeat the previous steps for all possible frontends;
6. If the account A is in the community, and the auditor considers it should not be, ask the controller why;
7. If the application does not request the consent of A to join the community, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the account A is not in the community AND the application requests the consent of A to join the community, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_3.a `basic`

The messenger MAY enable the user to configure who can add them to a community without their consent.

Prerequisite

- For each available frontend:

- A populated account , A, connected on a device d_1;
- An account B who is a contact of A, connected on a device d_2;

Verification steps

1. With the account A, search for the feature to enable another user to add them to a community without requesting their consent;
2. Mark down if this feature exists;
3. With the account A, enable this feature for B;
4. Using the account B, create a new community C and add A to the community;
5. Using the account A, inspect if it is in the community C;
6. Mark down if A is in the community C;
7. Repeat the previous steps for all possible frontends;
8. If A is not in the community, and the auditor considers it should be, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the feature to enable another user to add them to a community without requesting their consent exists AND A is in the community, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_4 `basic`

New Members of a community SHOULD NOT be able to see previous members of a community.

Prerequisite

- For each available frontend:
  - Four accounts A, B, C, D that are not mutual contacts, each of them connected on a device;

Verification steps

1. With account A, create and join the community CO;
2. Join the community CO with account B and C;
3. For each account A, B, C, send one message to the community CO;
4. Leave the community with account B and C;
5. Join the community CO with account D;
6. Mark down if, from the account D, the information associated to the community includes any identifiers of accounts B or C;
7. Repeat the previous steps for all possible frontends;

8. If the information associated to the community includes one or more identifiers of accounts B and C, and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the information associated to the community seen by D does not include any identifiers of accounts B and C, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_5 `basic`

Messengers SHOULD NOT offer a feature to list recently left members of a community.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B that is not a contact of A and is not sharing a community with A, connected on a device d_2;

Verification steps

1. With the account B, join a community C to which A belongs;
2. Send a message to the community C with account B;
3. Leave the community C with account B;
4. With the account A search, within the information associated to the community C, for a feature to display the list of recently left members of the community;
5. Mark down if a feature to display the list of recently left members of the community exists;
6. Repeat the previous steps for all possible frontends;
7. If a feature to display the list of recently left members of the community exists and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, a feature to display the list of recently left members of the community does not exist, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer a feature to display the list of recently left members of a community, the following requirements are not applicable (NA): COMMUNITIES_5.a, COMMUNITIES_5.b.

COMMUNITIES_5.a `basic`

When leaving the community, users MUST be offered the choice not to appear in the list of recently left members of the community.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B that is sharing a community C with A, connected on a device d_2;

Verification steps

1. With the account A leave the community C;
2. Mark down if the choice not to appear in the list of recently left members of the C is offered to the user;
3. Select the option not to appear in the list of recently left members of the community;
4. With the account B, inspect the list of recently left members of the community C;
5. Mark down if A is in the list of recently left members of the community C;
6. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature communities OR if the messenger does not offer a feature to display the list of recently left members of a community the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option not to appear in the list of recently left members of the community is offered to the leaving user AND if A is not in the list of recently left members of the community. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_5.b `basic`

Users listed in the list of recently left members of a community SHOULD NOT be listed longer than 72h after leaving.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a community C with A, connected on a device d_2;

Verification steps

1. With the account A leave the community C;
2. If prompted, choose to appear in the list of recently left members;
3. Wait 72 hours;
4. With the account B, inspect the list of recently left members of the community C;

5. Mark down of A is in the list of recently left members of the community C;
6. Repeat the previous steps for all possible frontends;
7. If A is in the list of recently left members of the community, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature communities OR if the messenger does not offer a feature to display the list of recently left members of a community the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, A is not in the list of recently left members of the community, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_6 `basic`

The community chat message history SHOULD NOT be visible to new members.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B that is not a contact of A and is not sharing a community with A, connected on a device d_2;;

Verification steps

1. With the account B, join C a community to which A belongs;
2. With the account B, inspect the message history of C;
3. Mark down if, from the point of view of account B, one or more messages are visible in the community C;
4. Repeat the previous steps for all possible frontends;
5. If one or more messages are visible and the auditor considers they should not be, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, no message is visible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_7 `basic`

The visibility of a community chat message history to new members MAY be configurable by the community admin.

Prerequisite

- For each available frontend:
  - An account A that is admin of a community G, connected on a device d_1;
  - Two accounts B and C that are not in G, connected on a device d_2;

Verification steps

1. With the account A, search for the feature to configure the visibility of a community chat message history to new members for community G;
2. Mark down if the feature to configure the visibility of a community chat message history to new members exists;
3. With the account A disable the visibility of a community chat message history to new members in community G;
4. With the account A, send a message m_1 to G;
5. With the account B, join the community G;
6. Mark down if, from the point of view of account B, one ore more messages are visible in the community G;
7. With the account A, enable the visibility of a community chat message history to new members;
8. With the account A, send a message m_2 to G;
9. With the account C, join the community G;
10. Mark down if, from the point of view of account C, one ore more messages are visible in the community G;
11. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature communities OR does not feature community chat message history OR does not feature setting the visibility of community chat message history to new members OR the concept of community admin does not exist, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the feature to configure the visibility of a community chat message history to new members exists AND from the point of view of account B no messages are visible AND from the point of view of account C only message m_2 and not m_1 is visible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the feature to configure the visibility of a community chat message history to new members does not exist, the following requirements are not applicable (NA): COMMUNITIES_7.a.

COMMUNITIES_7.a `basic`

This configuration SHOULD initially be set so that the history is not visible to new members.

Prerequisite

- An account A that is admin of a community C, connected on a device d_1;

Verification steps

1. With the account A, create a community C;
2. With the account A search for the feature to configure the visibility of a community chat message history to new members;
3. Mark down if the visibility of a community chat message history to new members is disabled;
4. Repeat the previous steps for all possible frontends;
5. If the visibility of a community chat message history to new members is not disabled, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature communities OR does not feature community chat message history, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the visibility of a community chat message history to new members is disabled, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_8 `basic`

The Messenger MAY offer the feature to display a limited number of past community chat messages to a new user of a community.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B that is not a contact of A and is not sharing a community with A, connected on a device d_2;

Verification steps

1. With the account A, select a community G to which A belongs and search for the setting that allows to set a number of past community chat messages to new members of the community;
2. Mark down the maximum number of past messages that can be set to be visible to new members of the community C;
3. With the account A, set the maximum number to be visible to new members of the community G;
4. With the account A send 101 messages to the community C;
5. With the account B, join the community C and inspect the messages of the community C;
6. Mark down the number of messages of the community C visible to the account B;
7. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature communities OR does not feature community chat message history OR does not feature setting the visibility of community message history to new members, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the number of messages of the community chat C visible to the account B is equal or less than 100, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_9 `basic`

Upon leaving a community, the past messages of the user MAY be masked as messages from an anonymous past member instead of appearing with that users identifier.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a community C with A, connected on a device d_2;

Verification steps

1. With the account A, send a message on the community C
2. Leave the community C with account A;
3. Inspect the chat message history of the community C;
4. Mark down if for all messages authored by A in the community C their author is set to a default value;
5. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature communities OR does not feature community chat message history, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, all messages authored by A in the community C have their author set to a default value, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_10 `basic`

Messenger SHOULD offer the feature to exit a community without notifying the users of the community.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a community CO with A, connected on a device d_2;

Verification steps

1. With the account A, leave the community GO;
2. Mark down if A was prompted to leave the community without notifying the other users of the community;
3. Select not to notify other members of the community;
4. With the account B, inspect potential notifications and information associated to the community CO;
5. Mark down if B is notified that A has left the community CO;
6. Repeat the previous steps for all possible frontends;
7. If A is not promped to leave the community silently, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, A is prompted to leave the community CO silently AND B is not notified that A has left the community CO, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_11 `basic`

Users that are members of a group MUST be notified, when that group is added to a community.

Prerequisite

- For each available frontend:
  - A populated account , A, that is an admin of a group G, connected on a device d_1;
  - An account B that is in the group G, connected on a device d_2;

Verification steps

1. Using the account A, create a community C and add the group G to the community;
2. Using the account B, inspect the general interface and the panel of the group G;
3. Mark down if the account B was notified that G was added to a community exists;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature communities OR does not feature groups the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the account B was notified that the group was added to a community. Otherwise, the requirement is not fulfilled (FAIL).

COMMUNITIES_11.a `basic`

Users that are members of a group SHOULD be given the opportunity to leave the group before being added to the community.

Prerequisite

- For each available frontend:
  - A populated account , A, that is an admin of a group G, connected on a device d_1;
  - An account B that is in the group G, connected on a device d_2;

Verification steps

1. Using the account A, create a community C and add the group G to the community;
2. Using the account B, inspect the general interface and the panel of the group G;
3. Mark down if the option to leave the group before G is added to C is offered;
4. Mark down if the group G is in the community C;
5. Repeat the previous steps for all possible frontends;;
6. If the option to leave the group is not offered, and the auditor considers it should, ask the controller why;
7. If the group is in the community, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature communities OR does not feature groups the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to leave the group is offered before the group is added to the community AND the group is not in the community, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.17 Broadcast

BROADCAST_1 `basic`

The list of the recipients of a broadcast SHOULD NOT be visible to other users.

Prerequisite

- A populated account A;
- Accounts B and C;

Verification steps

1. With the account A send a bradcast message m_BC to B and C;
2. With the account B, inspect the available information on the broadcast m_BC;
3. Mark down if the list of the recipients of the broadcast m_BC is visible to B;

4. Repeat the previous steps for all possible frontends;

5. If one or more recipient of $m_B C$ are visible from account B, and the auditor considers they should not be, ask the controller why;

Validation

If the messenger does not feature broadcasts, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, no recipient of broadcast $m_B C$ is visible from the account B, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.18 Channels

CHANNELS_1 `basic`

A users personal information SHOULD NOT be visible to other users in this channel who do not already know the users personal information.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - an account B that is not a contact of A, connected on a device d_2;

Verification steps

1. Select one channel C to which A belongs;
2. With the account B, join the channel G;
3. With the account B, inspect the user A and mark down the available information on the account A;
4. With the account A, inspect the user B and mark down the available information on the account B;
5. Repeat the previous steps for all possible frontends;
6. If one or more personal data of account A are visible from account B, and the auditor considers they should not be, ask the controller why;
7. If one or more personal data of account B are visible from account A, and the auditor considers they should not be, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, no personal data (other than the username) of the account A is visible from the account B AND no personal data (other than the username) of the account B is visible from the account A, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_2 `basic`

The list of the simple members of a channel SHOULD NOT be visible to other users.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - an account B that is not a contact of A, connected on a device d_2;

Verification steps

1. With the account A, select one channel C, and join it as a simple member;
2. With the account B, inspect the information on the channel C and mark down if A appears as a member of the channel;

3. Repeat the previous steps for all possible frontends;
4. If A appears as a member of the channel, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature channels, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, A does not appear as a member of the channel, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_3 `basic`

The application SHOULD feature an option for the user to prevent other users from adding them to a channel.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - an account B who is a contact of A, connected on a device d_2;

Verification steps

1. Using the account A, search for the setting that allows configuring if the user can be added to channels by another users;
2. Mark down if this setting exists;
3. Using the account A and this setting, disallow B to add A to channels;
4. Using the account B, create a new channel C, and attempt to add A to the channel C;
5. Mark down if A is added to C;
6. Repeat the previous steps for all possible frontends;
7. If the setting to configure if the user can be added to a channel by another user does not exist and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature channels, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the setting to configure if the user can be added to a channel by another user exists AND A could not be added to the channel C, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the option to configure if the user can be added to communities by another user does not exist AND this requirement is fulfilled (PASS), the following requirements are not applicable (NA): CHANNELS_3.a.

CHANNELS_3.a `basic`

By default, the option to configure if another user can add them to a channel SHOULD be disabled for all other users.

Prerequisite

- An account A connected on a device d_1;

Verification steps

1. Create a new account B and add A as a contact;
2. Using the account B, search for the setting to configure if the user can be added to channels by another user;
3. Mark down the current setting;
4. Using the account A, attempt to add B to a channel C;
5. Mark down if B is added to C;
6. Repeat the previous steps for all possible frontends;
7. If the default setting to configure if the user can be added to channels by another user allows adding the user in any tested frontend; and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature channels, OR if the messenger does not feature the option to configure if a user can be added to channels by another user AND CHANNELS_3 is fulfilled (PASS), the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the default setting to configure if the user can be added to channels by another user is set so that no other user is able to add the tested user to a channel AND B was not added to C, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_4 `basic`

Prior to being added to a channel the user SHOULD be asked for consent.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B who is a contact of A, connected on a device d_2;

Verification steps

1. With the account B, create a new channel C and add A to the channel;
2. Using the account A, inspect if it is in the channel C;
3. Mark down if the application requests the consent of A to join the channel;

4. Repeat the previous steps for all possible frontends;
5. If the account A is in the channel, and the auditor considers it should not be, ask the controller why;
6. If the application does not request the consent of A to join the channel, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature channels, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the account A is not in the channel AND the application requests the consent of A to join the channel, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_4.a `basic`

The messenger MAY enable the user to configure who can add them to a channel without their consent.

Prerequisite

- For each available frontend:
  - A populated account , A, connected on a device d_1;
  - An account B who is a contact of A, connected on a device d_2;

Verification steps

1. With the account A, search for the feature to enable another user to add them to a channel without requesting their consent;
2. Mark down if this feature exists;
3. With the account A, enable this feature for B;
4. Using the account B, create a new channel C and add A to the channel;
5. Using the account A, inspect if it is in the channel C;
6. Repeat the previous steps for all possible frontends;
7. If A is not in the channel, and the auditor considers it should be, ask the controller why;

Validation

If the messenger does not feature channels, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the feature to enable another user to add them to a channel without requesting their consent exists AND A is in the channel, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_5 `basic`

New members of a channel SHOULD NOT be able to see previous members of a channel.

Prerequisite

- For each available frontend:
  - Four accounts A, B, C, D that are not mutual contacts, each of them connected on a device.

Verification steps

1. With account A, create and join the channel CH;
2. Join the channel CH with account B and C;
3. For each account A, B, C, send one message to the channel CH;
4. Leave the channel with account B and C;
5. Join the channel CH with account D;
6. Mark down if, from the account D, the information associated to the channel includes any identifiers of accounts B and C;
7. Repeat the previous steps for all possible frontends;
8. If the information associated to the channel includes one or more identifiers of accounts B or C, and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature channels, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the information associated to the channel seen by D does not include any identifiers of accounts B and C, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_6 `basic`

The messenger SHOULD NOT offer a feature to display the list of recently left members of a channel.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is not a contact of A and is not sharing a channel with A, connected on a device d_2;

Verification steps

1. With the account B, join a channel C to which A belongs;
2. Send a message to the channel C with account B;
3. Leave the channel C with account B;

4. With the account A search, within the information associated to the channel G, for a feature to display the list of recently left members of the channel;
5. Mark down if a feature to display the list of recently left members of the channel exists;
6. Repeat the previous steps for all possible frontends;
7. If a feature to display the list of recently left members of the channel exists and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature communities, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, a feature to display the list of recently left members of the channel does not exist, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer a feature to display the list of recently left members of a channel, the following requirements are not applicable (NA): CHANNELS_6.a, CHANNELS_6.b.

CHANNELS_6.a `basic`

When leaving the channel, users MUST be offered the choice not to appear in the list of recently left members of the channel.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a channel C with A, connected on a device d_2;

Verification steps

1. With the account A leave the channel C;
2. Mark down if the choice not to appear in the list of recently left members of C is offered to the user;
3. Select the option not to appear in the list of recently left members of the channel;
4. With the account B, inspect the list of recently left members of the channel C;
5. Mark down if A is in the list of recently left members of the channel C;
6. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature channels OR if the messenger does not offer a feature to display the list of recently left members of a channel the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option not to appear in the list of recently left members of the channel is offered to the leaving user AND if A is not in the list of recently left members of the channel. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_6.b `basic`

Users listed in the list of recently left members of a channel SHOULD NOT be listed longer than 72h after leaving.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a channel C with A, connected on a device d_2;

Verification steps

1. With the account A leave the channel C;
2. If prompted, choose to appear in the list of recently left members;
3. Wait 72 hours;
4. With the account B, inspect the list of recently left members of the channel C;
5. Mark down of A is in the list of recently left members of the channel C;
6. Repeat the previous steps for all possible frontends;
7. If A is in the list of recently left members of the channel, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature channels OR if the messenger does not offer a feature to display the list of recently left members of a channel the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, A is not in the list of recently left members of the channel, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_7 `basic`

The channels message history SHOULD NOT be visible to new members.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is not a contact of A and is not sharing a channel with A, connected on a device d_2;

Verification steps

1. With the account B, join C a channel to which A belongs;
2. With the account B, inspect the message history of C;
3. Mark down if, from the point of view of account B, one or more messages are visible in the channel C;

4. Repeat the previous steps for all possible frontends;
5. If one or more messages are visible and the auditor considers they should not be, ask the controller why;

Validation

If the messenger does not feature channels, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, no message is visible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_8 `basic`

The visibility of a channels message history to new members MAY be configurable by the channel admin.

Prerequisite

- For each available frontend:
  - An account A that is admin of a channel CH, connected on a device d_1;
  - Two accounts B and C that are not in CH, connected on a device d_2;

Verification steps

1. With the account A, search for the feature to configure the visibility of a channel chat message history to new members for channel CH;
2. Mark down if the feature to configure the visibility of a channel message history to new members exists;
3. With the account A disable the visibility of a channel message history to new members in channel CH;
4. With the account A, send a message m_1 to CH;
5. With the account B, join the channel CH;
6. Mark down if, from the point of view of account B, one ore more messages are visible in the channel CH;
7. With the account A, enable the visibility of a channel message history to new members;
8. With the account A, send a message m_2 to CH;
9. With the account C, join the channel CH;
10. Mark down if, from the point of view of account C, one ore more messages are visible in the channel CH;
11. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature communities OR does not feature channel message history OR does not feature setting the visibility of channel message history to new members OR the concept of

channel admin does not exist, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the feature to configure the visibility of a channel message history to new members exists AND from the point of view of account B no messages are visible AND from the point of view of account C only message m_2 and not m_1 is visible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the feature to configure the visibility of a channel message history to new members does not exist, the following requirements are not applicable (NA): CHANNELS_8.a.

CHANNELS_8.a `basic`

This configuration SHOULD initially be set so that the history is not visible to new members.

Prerequisite

- An account A that is admin of a channel C, connected on a device d_1;

Verification steps

1. With the account A, create a channel C;
2. With the account A search for the feature to configure the visibility of a channel message history to new members;
3. Mark down if the visibility of a channel message history to new members is disabled;
4. Repeat the previous steps for all possible frontends;
5. If the visibility of a channel message history to new members is not disabled, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature channels OR does not feature channel message history, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the visibility of a channel message history to new members is disabled by default, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_9 `basic`

The Messenger MAY offer the feature to display a limited number of past channel messages to a new user of a channel.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is not a contact of A and is not sharing a channel with A, connected on a device d_2;

Verification steps

1. With the account A, select a channel C to which A belongs and search for the setting that allows to set a number of past channel messages to new members of the channel;
2. Mark down the maximum number of past messages that can be set to be visible to new members of the channel C;
3. With the account A, set the maximum number to be visible to new members of the channel C;
4. With the account A send 101 messages to the channel C;
5. With the account B, join the channel C and inspect the messages of the channel C;
6. Mark down the number of messages of the channel C visible to the account B;
7. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature channels OR does not feature channel message history OR does not feature setting the visibility of channel message history to new members, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the number of messages of the channel C visible to the account B is equal or less than 100, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_10 `basic`

Upon leaving a channel, the past messages of the user MAY be masked as messages from an anonymous past member instead of appearing with that users identifier.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a channel C with A, connected on a device d_2;

Verification steps

1. With the account A, send a message on the channel C;
2. Leave the channel C with account A;
3. With the account B, inspect the chat message history of the channel C;
4. Mark down if for all messages authored by A in the channel C their author is set to a default value;
5. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature channels OR does not feature channel message history, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available

frontends, all messages authored by A in the channel C have their author set to a default value, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_11 `basic`

The messenger SHOULD offer the possibility to send messages in the name of the channel instead of with a users own identifier.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a channel CH with A, connected on a device d_2;

Verification steps

1. With the account A, search for an option to send messages in the name of the channel;
2. Mark down if this option exists;
3. With the account A, send a message in the name of the channel;
4. With the account B, inspect the messages of the channel;
5. Mark down if the message sent by A appears as being sent by the channel and not by A;
6. Repeat the previous steps for all possible frontends;
7. If the option to send messages in the name of the channel does not exist, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature channels, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to send messages in the name of the channel exists AND the message sent by A appears as being sent by the channel and not by A, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CHANNELS_12 `basic`

Messenger SHOULD offer the feature to exit a channel without notifying the users of the channel.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is sharing a channel CH with A, connected on a device d_2;

Verification steps

1. With the account A, leave the channel GH;

2. Mark down if A was prompted to leave the channel without notifying the other users of the channel;
3. Select not to notify other members of the channel;
4. With the account B, inspect potential notifications and information associated to the channel CH;
5. Mark down if B is notified that A has left the channel CH;
6. Repeat the previous steps for all possible frontends;
7. If A is not promped to leave the channel silently, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature channels, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, A is prompted to leave the channel CH silently AND B is not notified that A has left the channel CH, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.19 Stories

STORY_1 `basic`

The messenger SHOULD enable the user to configure who can see their stories.

Prerequisite

- For each available frontend:
  - A populated account A, connected on a device d_1;
  - An account B that is a contact of A, connected on a device d_2;

Verification steps

1. Using the account A, search for the setting to configure who can see the stories authored by A;
2. Mark down if this setting exists;
3. Using the account A, create a story and configure the setting such that B cannot see the story;
4. Using the account B, inspect the stories of A and try to see the story that A just created;
5. Mark down if the story is visible by B;
6. Repeat the previous steps for all possible frontends;
7. If the setting to configure who can see the stories does not exist and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature stories, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the setting to configure who can see the stories exists AND the story is not visible by B, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the feature to configure who can see stories AND this requirement is fulfilled (PASS), the following requirements are not applicable (NA): STORY_1.a.

STORY_1.a `basic`

This configuration SHOULD initially be set to the most privacy preserving setting.

Prerequisite

- a device for each available frontend;

Verification steps

1. Create an account A;
2. Inspect the setting to configure who can see the stories;
3. Mark down if it is set so that no other user can see the stories;
4. Repeat the previous steps for all possible frontends;
5. If the setting is set so that other users can see the stories and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature stories, OR an option to configure the visibility of user generated stories AND STORY_1 is fulfilled (PASS), the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the setting is set so that no other user can see the stories, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.9.20 Spellchecks

SPELLCHECK_1 `basic`

The spellchecking SHOULD be performed entirely locally.

Prerequisite

- None;

Verification steps

1. Ask the controller if the spellchecking components send data to a remote entity;
2. Verify if the spellchecking components send data to a remote entity;

3. Repeat the previous steps for all possible frontends;
4. If any spellchecking component sends data to a remote entity, and the auditor considers they should not, ask the controller why;.

Validation

If the messenger does not offer a spellchecking feature the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, none of the spellchecking components send data to a remote entity, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SPELLCHECK_2 `basic`

An external third party spellchecking feature MUST be configurable by the user.

Prerequisite

- an account, A.

Verification steps

1. Login with the account A;
2. Search for the feature to configure the external third party spellchecking feature;
3. Verify that the external third party spellchecking feature can be configured by the user;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not offer a spellchecking feature the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the external third party spellchecking feature can be configured by the user. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not offer an external third party spellchecking feature the following requirements are not applicable (NA): SPELLCHECK_2.a, SPELLCHECK_3, SPELLCHECK_4.

SPELLCHECK_2.a `basic`

An external third party spellchecking feature MUST be disabled by default.

Prerequisite

- None;

Verification steps

1. Create an account;
2. Inspect the configuration of the external third party spellchecking service;

3. Verify that the external third party spellchecking service is disabled;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not offer an external third party spellchecking feature the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the external third party spellchecking feature is disabled. Otherwise, the requirement is not fulfilled (FAIL).

SPELLCHECK_3 `basic`

Before activating the external third party spellchecking feature the messenger MUST inform the user of the associated risks.

Prerequisite

- None;

Verification steps

1. Create an account;
2. Activate the external third party spellchecking feature;
3. Verify that the user is informed of the risks associated with an external third party spellchecking feature;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not offer an external third party spellchecking feature the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user is informed of the risks. Otherwise, the requirement is not fulfilled (FAIL).

SPELLCHECK_4 `basic`

The controller MUST provide a valid legal basis for performing spellchecking through an external third party.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the legal basis for performing spellchecking through an external third party;
2. Verify that the legal basis is valid;
3. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not offer an external third party spellchecking feature the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the legal basis is valid. Otherwise, the requirement is not fulfilled (FAIL).

2.9.21 Keyboard

KEYBOARD_1 `basic`

The messenger SHOULD check if the input method is a third party keyboard.

Prerequisite

- None;

Verification steps

1. Ask the controller if the application checks if the input method is a third party keyboard;
2. Repeat the previous steps for all possible frontends;
3. If the application does not check if the input method is a third party keyboard, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application checks if the input method is a third party keyboard, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the messenger does not check if the input method is a third party keyboard and this requirement is fulfilled (PASS), the following requirements are not applicable (NA): KEYBOARD_1.a.

KEYBOARD_1.a `basic`

If a third party keyboard is detected, the messenger SHOULD inform the user of the associated risks.

Prerequisite

- None;

Verification steps

1. Ask the controller if the user is informed of associated risks in case a third-party keyboard is detected;
2. Mark down if the user is informed of associated risks in case a third party keyboard is detected;
3. Repeat the previous steps for all possible frontends;;
4. If the user is not informed of associated risks in case a third party keyboard is detected, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature a check if the input method is a third party keyboard and the requirement KEYBOARD_1 is fulfilled (PASS), the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user is informed of associated risks in case a third party keyboard is detected, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

KEYBOARD_1.b `basic`

The messenger application SHOULD offer the user the usage of a local application keyboard.

Prerequisite

- An account for each available frontend;

Verification steps

1. Search for the keyboard-related setting of the application and inspect the available options;
2. Mark down if the option to use a local keyboard is offered. Such keyboard includes, but is not limited to the keyboard provided by the operating system or the keyboard provided by default by the device manufacturer;
3. Repeat the previous steps for all possible frontends;
4. If the option to use a local keyboard is not offered, and the auditor considers it should, ask the controller why;;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the option to use a local keyboard is offered, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. . Otherwise, the requirement is not fulfilled (FAIL).

KEYBOARD_2 `basic`

The messenger SHOULD offer the option to signal the keyboard to disable any data collection beyond what is absolutely necessary to perform the basic input processing, including but not limited to word predictions or model training.

Prerequisite

- an account A;

Verification steps

1. Login with the account A;
2. Search for the option to signal the keyboard to disable any data collection;
3. Verify that the option to signal the keyboard to disable any data collection exists;
4. Repeat the previous steps for all possible frontends;

5. If the option to signal the keyboard to disable any data collection does not exist, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the option to signal the keyboard to disable any data collection exists, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the option to signal the keyboard to disable any data collection does not exist AND this requirement is fulfilled, the following requirements are not applicable (NA): KEYBOARD_2.a.

KEYBOARD_2.a `basic`

The option to signal the keyboard to disable any data collection SHOULD be enabled by default.

Prerequisite

- None;

Verification steps

1. Create an account;
2. Inspect the option to signal the keyboard to disable any data collection;
3. Verify that the option to signal the keyboard to disable any data collection is enabled;
4. Repeat the previous steps for all possible frontends;

Validation

If the option to signal the keyboard to disable any data collection does not exist AND KEYBOARD_2 is fulfilled the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to signal the keyboard to disable any data collection is enabled. Otherwise, the requirement is not fulfilled (FAIL).

2.9.22 Message translation

TRANSL_1 `basic`

The external third party translation feature MUST be configurable by the user.

Prerequisite

- A populated account A;

Verification steps

1. Search for the option to disable or enable external third party translation feature in the settings panel;
2. Mark down if the option to disable or enable external third party translation feature exists;

3. Disable the external third party translation feature;
4. Select one received message and try to translate it to any available language;
5. Mark down if the message is translated;
6. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not include an external third party translation feature the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to disable or enable external third party translation feature exists AND the message is not translated. Otherwise, the requirement is not fulfilled (FAIL).

TRANSL_1.a `basic`

The external third party translation feature MUST be disabled by default.

Prerequisite

- None;

Verification steps

1. Create an account A;
2. Inspect the option to configure the external third party translation feature;
3. Mark down if the external third party translation feature is disabled;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not include an external third party translation feature the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the external third party translation feature is disabled. Otherwise, the requirement is not fulfilled (FAIL).

TRANSL_2 `basic`

Before activating the external third party translation feature the messenger MUST inform the user of the associated risks.

Prerequisite

- None;

Verification steps

1. Create an account A;
2. Enable the external third party translation feature;
3. Mark down if the user was informed of the risks associated with an external third party translation feature before enabling the feature;

4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not include an external third party translation feature the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user is informed of the risks. Otherwise, the requirement is not fulfilled (FAIL).

2.9.23 Password protected conversations

PRIV_CHAT_1 basic

The password SHOULD be requested and verified every time the user opens a password protected conversation.

Prerequisite

- A populated account A;

Verification steps

1. Login with the account A;
2. Create a password protected discussion D with a contact of A;
3. For each of the listed scenarios repeat the given procedure; Scenarios:
   - Coming from another discussion;
   - Putting the application in the background;
   - Closing the application;

   Procedure:
   (a) Try to access the protected discussion D;
   (b) Mark down if the password is requested;
   (c) Try entering a wrong password;
   (d) Mark down if access is granted;
   (e) Try entering the correct password;
   (f) Mark down if access is granted;
4. Repeat the previous steps for all possible frontends;
5. If the password is not requested, and the auditor considers it should be, ask the controller why;
6. If the access is granted when providing the incorrect password, and the auditor considers it should be, ask the controller why;

Validation

If the messenger does not offer password protected discussions the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the password is always requested AND the access is not granted when providing an incorrect password. Otherwise, the requirement is not fulfilled (FAIL).

2.9.24 Direct communications

DIRECT_COM_1 basic

Direct communications MUST be configurable by the user.

Prerequisite

- A populated account A connected on a device d_1;

Verification steps

1. Search for the option to disable or enable external direct communications;
2. Mark down if the option to disable or enable external direct communications exists;
3. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature direct communications the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to disable or enable external direct communications exists. Otherwise, the requirement is not fulfilled (FAIL).

DIRECT_COM_1.a `basic`

Direct communications MUST be disabled by default.

Prerequisite

- A device for each available frontend;

Verification steps

1. Create an account A;
2. Inspect the option to configure direct communications;
3. Mark down if direct communications are disabled;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature direct communications the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, direct communications are disabled. Otherwise, the requirement is not fulfilled (FAIL).

DIRECT_COM_1.b `basic`

The user MUST be informed of the risk of using direct communications before activating the feature.

Prerequisite

- A device for each available frontend;

Verification steps

1. Create an account A;
2. Enable direct communications;
3. Mark down if the user was informed of the risk of using direct communications before enabling the feature;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature direct communications the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user was informed of the risk of using direct communications before enabling the feature. Otherwise, the requirement is not fulfilled (FAIL).

2.9.25 Network proxy

PROXY_1 `basic`

The network proxy feature MUST be configurable by the user.

Prerequisite

- A populated account A connected on a device d_1;

Verification steps

1. Search for the option to disable or enable the network proxy;
2. Mark down if the option to disable or enable the network proxy exists;
3. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature a network proxy, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the option to disable or enable external the network proxy exists. Otherwise, the requirement is not fulfilled (FAIL).

PROXY_1.a `basic`

The network proxy feature MUST be disabled by default.

Prerequisite

- A device for each available frontend;

Verification steps

1. Create an account A;
2. Inspect the option to configure the network proxy;
3. Mark down if the network proxy is disabled;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature a network proxy, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the network proxy is disabled. Otherwise, the requirement is not fulfilled (FAIL).

PROXY_1.b basic

The user MUST be informed of the risk of using a network proxy before activating the feature.

Prerequisite

- A device for each available frontend;

Verification steps

1. Create an account A;
2. Enable the network proxy;
3. Mark down if the user was informed of the risk of using network proxies before enabling the feature;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature a network proxy, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user was informed of the risk of using network proxies before enabling the feature. Otherwise, the requirement is not fulfilled (FAIL).

2.9.26 Data collection for analytics and crash reports

ANALYTICS_1 basic

Analytics and debugging data collection MUST be configurable by the user.

Prerequisite

- An account A;

Verification steps

1. Search for the option to enable or disable data collection for analytics purposes;
2. Mark down if the option to enable or disable data collection for analytics purposes exists;
3. Search for the option to enable or disable data collection for debugging purposes;
4. Mark down if the option to enable or disable data collection for debugging purposes exists;
5. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature data collection for analytics and crash reports, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, option to enable or disable data collection for analytics purposes exists AND the option to enable or disable data collection for debugging purposes exists. Otherwise, the requirement is not fulfilled (FAIL).

ANALYTICS_1.a `basic`

Analytics and debugging data collection SHOULD be disabled by default.

Prerequisite

- A device for each available frontend.

Verification steps

1. Create an account A and connect with the device;
2. Inspect the options to configure analytics and debugging data collection;
3. Mark down if the data collection for analytics purposes is disabled;
4. Mark down if the data collection for debugging purposes is disabled;
5. Repeat the previous steps for all possible frontends;
6. If the data collection for analytics purposes is enabled, and the auditor considers it should not be, ask the controller why;
7. If the data collection for debugging purposes is enabled, and the auditor considers it should not be, ask the controller why;

Validation

If the messenger does not feature data collection for analytics and debugging, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the data collection for analytics purposes is disabled AND the data collection for debugging purposes is disabled, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

ANALYTICS_2 `basic`

The user MUST be informed of the risk of enabling analytics and debugging data collection before activating the feature.

Prerequisite

- An account A;
- A device for each available frontend;

Verification steps

1. Disable the analytics and debugging data collection features;
2. Enable the analytics data collection feature;
3. Mark down if the user was informed of the risks of analytics data collection before activating the feature;
4. Enable the debugging data collection feature;
5. Mark down if the user was informed of the risks of debugging data collection before activating the feature;

6. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature data collection for analytics and debugging the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the user was informed of the risks of analytics data collection before activating the feature AND the user was informed of the risks of debugging data collection before activating the feature. Otherwise, the requirement is not fulfilled (FAIL).

ANALYTICS_3 `basic`

Error messages and notifications MUST NOT contain personal data or other sensitive data (such as a keys or authentication token). [fSidI20, O.Source_3]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all possible error messages and notifications and the corresponding dataset of the application that is included in the crash report;
2. Mark down if any of the corresponding datasets of the applications crash reports access profile data, personal data or other sensitive data of the user;
3. Ask the controller which measures are taken to prevent personal data or other sensitive data to be included in error messages and notifications;
4. Repeat the previous steps for all possible frontends;

Validation

If the messenger does not feature data collection for analytics and debugging, the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the auditor came to the conclusion that no personal data or other sensitive data is included in the error messages and notifications. Otherwise, the requirement is not fulfilled (FAIL).

2.10 Processor (Art. 28 GDPR)

PROCESSOR_1 `basic`

The messenger controller SHOULD use processor (third party services / libraries processing personal data) providing sufficient guarantees to implement appropriate technical and organisational measures in such a manner that processing will meet the requirements of the GDPR and ensure the protection of the rights of the data subject.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of the processor associated to the messenger;
2. Ask the controller to provide, for each processor, evidences proving the compliance of the processor with the GDPR;
3. Verify that the evidences provided prove the compliance of all processors with the GDPR;
4. Repeat the previous steps for all possible frontends;
5. If for one or more processors the provided evidences are not sufficient to prove GDPR compliance, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all processor, the controller provided evidences proving compliance with GDPR. Otherwise, the requirement is not fulfilled (FAIL).

2.11 Security of processing (Art. 32 GDPR)

2.11.1 General security requirements

SECURITY_1 `basic`

The controller MUST implement appropriate technical and organisational measures to ensure a level of security appropriate to the risk.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of identified risks;
2. Verify that the provided list of risks is plausible;
3. Ask the controller to provide, for each listed risk, a list of technical and organisational measures implemented to eliminate or mitigate the risk.
4. Verify that all risks are met with appropriate technical and organisational measures;
5. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the provided list of risks is plausible AND all listed risks are met with appropriate technical and organisational measures. Otherwise, the requirement is not fulfilled (FAIL).

SECURITY_2 `basic`

Personal data SHOULD be encrypted at rest.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of personal data used by the application;
2. Ask the controller to provide, for each personal data, if it is encrypted when at rest on the device;
3. Mark down if all personal data are encrypted when at rest on the device;
4. Repeat the previous steps for all possible frontends;
5. If one or more personal data are not encrypted when at rest on the device, and the auditor considers they should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all personal data at rest on the device are encrypted, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SECURITY_3 `basic`

Personal data SHOULD be encrypted in transit.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of personal data transmitted to and from the application;
2. Ask the controller to provide, for each personal data, if it is encrypted when in transit;
3. Mark down if all personal data are encrypted when in transit;
4. Repeat the previous steps for all possible frontends;
5. If one or more personal data is not encrypted when in transit, and the auditor considers they should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all personal data are encrypted when in transit, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SECURITY_4 `basic`

Appropriate technical measures SHOULD be implemented to prevent unauthorized access to personal data processed by the application.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all personal data used by the application;
2. Ask the controller to provide, for each personal data, the technical measures to prevent unauthorized access to the data;
3. Verify for each personal data that technical measures are implemented to prevent unauthorized access;
4. Repeat the previous steps for all possible frontends;
5. If for one or more personal data, no technical measure is implemented to prevent unauthorized access, and the auditor considers it should, ask the controller why;.

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all personal data processed by the application appropriate technical measures are implemented to prevent unauthorized access. Otherwise, the requirement is not fulfilled (FAIL).

2.11.2 Security of communication

COMM_1 `basic`

Personal data transmitted from and to the application MUST be done through a secure communication channel.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all communication channels carrying personal data to and from the application, along with the measures to secure these channels;
2. Verify that each communication channel carrying personal data is secured;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all communication channels carrying personal data are secured. Otherwise, the requirement is not fulfilled (FAIL).

COMM_1 `intermediate`

Personal data transmitted from and to the application MUST be done through a secure communication channel.

Prerequisite

- A passive network traffic capture environment (see B.2.1)
- An end device

Verification steps

1. Connect the device the network capture environment and start the capture;
2. Create an account (A.1.1) and login with the account;
3. Populate the account (A.1.4);
4. Proceed to standard usage (A.1.5);
5. Stop the capture;
6. Identify and inspect all communication that are not secured;
7. Verify that no personal information is included in the communications that are not secured. To this aim, the auditor search for personal data artifacts in the unsecured communication. Personal data artifacts include, but are not limited to:
   - Name of personal data: email, name, firstname, lastname, IEMI, number ...
   - Values of personal data: "johndoe@mail.com", "John Doe", "John", "Doe", "107738152789381", "+33060504030201" ...

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no personal information is included in the captured communications that are not secured. Otherwise, the requirement is not fulfilled (FAIL).

COMM_2 `basic`

Data transmitted from and to the application SHOULD be done through a secure communication channel.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all communication channels carrying any data to and from the application, along with the measures to secure these channels;
2. Verify that each communication channel is secured;
3. Repeat the previous steps for all possible frontends;
4. If one or more channel are not secured and and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all communication channels are secured. Otherwise, the requirement is not fulfilled (FAIL).

COMM_3 `basic`

Secure communication channels MUST NOT be implemented with obsolete or deprecated protocols.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all secured communication channels to and from the application, along with the details of the protocols used to secure them;
2. Verify that protocols used to secure communications are not obsolete or deprecated. The auditor can consult national guidelines or resources such as the following to assess this compliance:
   - ANSSI's Security Recommendations for TLS [ANS17]
   - NIST's Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations [NIS19]
   - ENISA's Study on cryptographic protocols [ENI14b]
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, protocols used to secure communications are not obsolete or deprecated. Otherwise, the requirement is not fulfilled (FAIL).

COMM_4 `basic`

Deprecated cipher suites MUST be disabled.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all enabled cipher suites;
2. Verify that none of the enabled cipher suites is deprecated. The auditor can consult national guidelines or resources such as the following to assess this compliance:
   - ANSSI's Security Recommendations for TLS [ANS17]
   - NIST's Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations
   - IANA's Transport Layer Security (TLS) Parameters [IAN23]
   - ENISA's Algorithms, Key Sizes and Parameters Report[ENI14a]
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, none of the enabled cipher suites are deprecated. Otherwise, the requirement is not fulfilled (FAIL).

COMM_5 `basic`

The application SHOULD only use implementations of secure network protocols that are coming from reliable sources and that are maintained.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of the used implementations and libraries of the secure network protocols used by the application;
2. Verify, for each implementation, that it is coming from a reliable source;
3. Verify, for each implementation, that it is maintained;
4. Repeat the previous steps for all possible frontends;
5. If one or more implementations are not coming from a reliable source, and the auditor considers they should be, ask the controller why;
6. If one or more implementations are not maintained, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all implementations of secure network protocols are coming from a reliable sources AND all implementation of secure network protocols are maintained, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

COMM_6 `basic`

The application SHOULD NOT use an implementation of secure network protocols internally developed by the controller.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of the used implementations and libraries of the secure network protocols used by the application;
2. Verify, for each implementation, that it is not internally developed by the controller;
3. Ask the controller, for each implementation, if the implementation of secure network protocol has been modified by the controller;
4. Repeat the previous steps for all possible frontends;
5. If one or more implementations is internally developed by the controller, and the auditor considers they should not be, ask the controller why;

6. If one or more implementations has been modified by the controller, and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no implementation of secure network protocols is internally developed by the controller AND no implementation of secure network protocols has been modified by the controller, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.3 End-to-end encryption

E2E_ENC_1 `basic`

The messages SHOULD be end-to-end encrypted.

Prerequisite

- None;

Verification steps

1. Ask the controller if all the messages are end-to-end encrypted, and if applicable, ask the controller to provide details of the end-to-end encryption scheme;
2. Verify that all the messages are end-to-end encrypted;
3. Ask the controller if the content of end-to-end encrypted messages are sent to another entity than the intended recipient. This includes, but is not limited to, a server performing caching or antivirus analysis;
4. Verify that the content of end-to-end encrypted messages are not sent to another entity than the intended recipient;
5. Ask the controller who knows the key capable of decrypting the message;
6. Verify that only the intended recipient knows the key capable of decrypting an end-to-end encrypted message.
7. Repeat the previous steps for all possible frontends;
8. If one or more messages are not end-to-end encrypted, and the auditor considers they should be, ask the controller why;
9. If the content of end-to-end encrypted messages is sent to another entity other than the intended recipient, and the auditor considers it should not be, ask the controller why;
10. If the key capable of decrypting an end-to-end encrypted message is known by another entity than the intended recipient, and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all messages are end-to-end encrypted AND the content of end-to-end encrypted messages is not sent to another entity other than the

intended recipient AND the key capable of decrypting an end-to-end encrypted message is only known by the intended recipient, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

E2E_ENC_2 `basic`

The application SHOULD offer a way to verify the public key of a contact.

Prerequisite

- A populated account A;
- An account B that is an unverified contact of A;

Verification steps

1. Login with the account A;
2. Verify that, for each contact of A, the application offers a way to verify the public key of the contact;
3. Select one unverified contact B and proceed to the public key verification;
   (a) Using a second device, login with the account B;
   (b) Follow the instruction for the mutual verification;
4. Verify that the public key verification was successful: the procedure finished without error and the contact B is now verified;
5. Repeat the previous steps for all possible frontends;
6. If for one or more contacts, a way to verify the public key is not offered, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all contacts, the application offers a way to verify the public key AND the public key verification was successful, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.4 Secure data transfer security for access right and portability

SEC_TRANS_1 `basic`

During the process of requesting data in the context of exercising the right to access (2.5), the controller SHOULD offer the user a way to provide an encryption key during the process of requesting the data.

Prerequisite

- A populated account, A;

Verification steps

1. Proceed to the access right procedure with the account A;
2. Markdown if the user is offered a way to provide an encryption key with which the requested data will be encrypted;
3. Complete the request for access and provide a key k;
4. Mark down if the received data package is encrypted with the key k;
5. Repeat the previous steps for all possible frontends;
6. If the user is not offered a way to provide an encryption key, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the user is offered a way to provide an encryption key AND the resulting data extract is encrypted with the provided key, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SEC_TRANS_2 `basic`

During the process of requesting data in the context of exercise of right to access (2.5), the controller SHOULD NOT offer the user a way to provide a freely chosen passphrase with which the data will be encrypted.

Prerequisite

- A populated account, A;

Verification steps

1. Proceed to the access right procedure with the account A;
2. Markdown if the controller offers a way to provide a freely chosen passphrase with which the data will be encrypted;
3. Repeat the previous steps for all possible frontends;
4. If the controller offers a way to provide a freely chosen passphrase, and the auditor considers it should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller does not offer a way to provide a freely chosen passphrase with which the data will be encrypted, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SEC_TRANS_3 `basic`

During the process of requesting data in the context of data portability (2.8), the controller SHOULD offer the user a way to provide an encryption key during the process of requesting the data.

Prerequisite

- A populated account, A.

Verification steps

1. Proceed to the data portability procedure with the account A;
2. Markdown if the user is offered a way to provide an encryption key with which the requested data will be encrypted;
3. Complete the request and provide a key k;
4. Mark down if the received data package is encrypted with the key k;
5. Repeat the previous steps for all possible frontends;
6. If the user is not offered a way to provide an encryption key, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the user is offered a way to provide an encryption key AND the resulting data extract is encrypted with the provided key, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SEC_TRANS_4 `basic`

During the process of requesting data in the context of data portability (2.8), the controller SHOULD NOT offer the user a way to provide a freely chosen passphrase with which the data will be encrypted.

Prerequisite

- A populated account, A.

Verification steps

1. Proceed to the data portability procedure with the account A;
2. Markdown if the controller offers a way to provide a freely chosen passphrase with which the data will be encrypted;
3. Repeat the previous steps for all possible frontends;;
4. If the controller offers a way to provide a freely chosen passphrase, and the auditor considers it should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller does not offer a way to provide a freely chosen passphrase with which the data will be encrypted, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.5 Certificates and trust anchors

CERT_1 `basic`

The validity of the certificates SHOULD be verified by the application.

Prerequisite

- None;

Verification steps

1. Ask the controller if the validity of certificates is verified within the application by checking:
   - Valid subject domain names;
   - Valid issue and expiry dates;
   - Valid signature by a trusted party;
2. Ask the controller to provide details on how an invalid certificate is handled by the application;
3. Repeat the previous steps for all possible frontends;
4. If for one or more certificates, its validity is not correctly verified, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all certificates their validity is correctly verified within the application AND the application refuses to accept invalid certificates, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CERT_1 `intermediate`

The validity of the certificates SHOULD be verified by the application.

Prerequisite

- A populated account connected on a device;
- A passive capture environment
- A client-side SSL auditing tool (e.g. qsslaudit[a])

Verification steps

1. Connect the device to the network capture environment;
2. Identify the hosts contacted by the application (A.1.7) while performing the following steps:
   (a) Create an account (A.1.1) and login with the account;
   (b) Populate the account (A.1.4);
   (c) Proceed to standard use (A.1.5);
3. For each host run the following test with the auditing tool while running and using the app;

(a) certificate trust test with www.example.com common name signed by user-supplied certificate

(b) certificate trust test with www.example.com common name signed by user-supplied CA certificate

4. Mark down the results of the tests;
5. Repeat the previous steps for all possible frontends;;
6. If one or more tests failed, and the auditor considers they should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all tests passed, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL)..

[a] https://github.com/gremwell/qsslcaudit

CERT_2 `basic`

The messenger SHOULD NOT use self-signed certificates.

Prerequisite

- None;

Verification steps

1. Ask the controller if the application accepts self-signed certificates;
2. Mark down if self-signed certificates are used or accepted by the application;
3. Repeat the previous steps for all possible frontends;
4. If the application accepts self-signed certificates, and the auditor considers it should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application does not accept self-signed certificates, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CERT_3 `basic`

The messenger MAY limit the list of trusted certificate authorities (CA) to a subset of entities considered trustworthy by the controller.

Prerequisite

- None;

Verification steps

1. Ask the controller if the application limits the set of trusted certification authorities (CA);
2. Mark down the answer;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application limits the set of trusted CA.

CERT_4 `basic`

The messenger SHOULD use certificate pinning.

Prerequisite

- None;

Verification steps

1. Ask the controller if the application uses certificate pinning;
2. Mark down if certificate pinning is used;
3. Repeat the previous steps for all possible frontends;
4. If the application does not use certificate pinning, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application uses certificate pinning, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.6 Key management and storage

KEY_MGT_1 `basic`

Cryptographic keys SHOULD be generated using a secure cryptographic random number generator.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all keys used in the application, along with how they are generated;
2. Verify, for each key, if it is generated using a secure cryptographic random number generator. Such sources include, but are not limited to:
   - Android: `Keystore.KeyPairGenerator`, `Keystore.KeyGenerator` or, `java.security.SecureRandom`

- iOS: `SecKeyCreateRandomKey`, `SecKeyCreateEncryptedData`, `SecRandomCopyBytes`
- Linux: `/dev/random` and `/dev/urandom`
- Windows: `CryptGenRandom`

3. Repeat the previous steps for all possible frontends;
4. If one or more random keys are not generated using a secure cryptographic random number generator, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all keys are generated using a secure cryptographic random number generator, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

KEY_MGT_2 `basic`

Cryptographic keys SHOULD NOT be used for more than one purpose. [fSidI20, O.Cryp_4]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all keys used in the application along with their purposes;
2. Verify, for each key, that it is not used for more than one purpose;
3. Repeat the previous steps for all possible frontends;
4. If one or more keys are used for more than one purpose, and the auditor considers it should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all keys are not use for more than one purpose, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

KEY_MGT_3 `basic`

Cryptographic keys SHOULD be stored in a secure environment.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all keys used in the application along with their storage location;
2. Verify, for each key, that it is stored in a secure environment. Secure environments include, but are not limited to:

- Android: `KeyStore`, `KeyChain`
- iOS: `KeyStore`, `Secure Enclave`

3. Repeat the previous steps for all possible frontends;
4. If one or more keys are not stored in a secure environment, and the auditor considers they should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all keys are stored in a secure environment, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If the cryptographic keys are not stored in a secure environment AND this requirement is fulfilled (PASS), the following requirements are not applicable (NA): KEY_MGT_3.a

KEY_MGT_3.a `basic`

Cryptographic keys SHOULD be stored in a secure hardware environment.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all keys used in the application along with their storage location;
2. Verify, for each key, that it is stored in a secure hardware environment. Secure hardware environments include, but are not limited to:
   - TPM (Trusted Platform Module)
   - TEE (Trusted Execution Environment)
3. Repeat the previous steps for all possible frontends;
4. If one or more keys are not stored in a secure hardware environment, and the auditor considers they should, ask the controller why;

Validation

If the cryptographic keys are not stored in a secure environment AND KEY_MGT_3 isfulfilled (PASS), the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, all keys are stored in a secure hardware environment, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

KEY_MGT_4 `basic`

Cryptographic keys SHOULD NOT be hardcoded.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all keys used in the application;
2. Ask the controller, for each key, if it is included in the code of the application;
3. Mark down if no key is included in the code of the application;
4. Repeat the previous steps for all possible frontends;
5. If one or more keys are included in the code of the application, and the auditor considers they should not be, ask the controller why;.

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no key is included in the code of the application, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.7 Cryptography

CRYPTO_1 `basic`

The messenger SHOULD use cryptographic primitives, schemes and key lengths corresponding to the state of the art.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all cryptographic primitives, schemes and key length used by the application, as well as their corresponding use;
2. Verify that each element is compliant with the state of the art. The auditor can consult national guidelines or resources such as the following to assess this compliance:
   - BSI TR-02102-1: Cryptographic Mechanisms: Recommendations and Key Lengths [fIS23]
   - Guide de sélection d'algorithmes cryptographiques (ANSSI) [ANS21a]
   - Algorithms, key size and parameters report 2014 (ENISA) [ENI14a]
3. Repeat the previous steps for all possible frontends;
4. If one or more primitives, schemes or key length do not correspond to the state of the art and and the auditor considers they should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the auditor came to the conclusion that the used primitives, schemes and key lengths are state of the art, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CRYPTO_2 `basic`

The messenger SHOULD NOT use obsolete cryptographic primitives and schemes.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all cryptographic primitives and schemes used by the application;
2. Verify that none of the used cryptographic primitives or schemes is obsolete. The auditor can consult national guidelines to assess the obsolete status of a primitive and schemes. Obsolete cryptographic primitives and schemes include but are not limited to:
   - DES, triple-DES, Kasumi, Blowfish;
   - SHA-1, SHA-2, RIPEMD-128, RIPEMD-160, MD5;
   - RC4, A5/1, A5/2, E0, Trivium, SNOW 2.0;
   - RSA-PKCS#1v1.5, RSA-FDH, ISO-9796-2 RSA-DS3, (EC)DSA,(EC)GDSA, (EC)KDSA,(EC)RDSA public key [ENI14a];
3. Repeat the previous steps for all possible frontends;
4. If one or more primitives or schemes used by the application are obsolete and the auditor considers they should not be used, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application does not use obsolete or deprecated primitive or scheme, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CRYPTO_2.a `basic`

The messenger SHOULD NOT use schemes with output, key lengths and parameters below the following thresholds :

- *Symmetric Ciphers: key shorter than 128 bits*
- *Hash function: output shorter than 256 bits*
- *MAC (HMAC [RFC2104, ISO9797-2]): output shorter than 128 bits*
- *RSA: modulus of shorter than 3072 bits, and exponent shorter than 2^{16} bits*

- *Finite Field Discrete Log problem (e.g. MODP [RFC3526]): modulus shorter than 3072 bits and*
- *Elliptic Curve Discrete Log problem: key shorter than 256 bits*
- *Pairing: key shorter than 3072 bits*

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all cryptographic primitives and schemes used by the application, along with their parameters, key lengths, and output lengths;
2. Verify that none of these values is below the following thresholds:
   - Symmetric Ciphers: key shorter than 128 bits
   - Hash function: output shorter than 256 bits
   - MAC (HMAC [RFC2104, ISO9797-2]): output shorter than 128 bits
   - RSA : modulus of shorter than 3072 bits, and exponent shorter than 2^{16} bits
   - Finite Field Discrete Log problem (e.g. MODP [RFC3526]): modulus shorter than 3072 bits and
   - Elliptic Curve Discrete Log problem: key shorter than 256 bits
   - Pairing: key shorter than 3072 bits
3. Repeat the previous steps for all possible frontends;
4. If one or more parameters, key lengths, or output length are below the listed threshold and the auditor considers they should not be used, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no obsolete or deprecated primitive or scheme has been identified, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CRYPTO_3 `basic`

The messenger SHOULD only use implementation of cryptographic primitives and schemes that are coming from reliable sources and that are maintained.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of the implementation of cryptographic primitives and schemes used by the application;
2. Verify, for each implementation, that it is coming from a reliable source;

3. Verify, for each implementation, that it is maintained;
4. Repeat the previous steps for all possible frontends;
5. If one or more implementations are not coming from a reliable source, and the auditor considers they should beask the controller why;
6. If one or more implementations are not maintained, and the auditor considers they should beask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, implementations of cryptographic primitives and schemes are maintained AND are coming from reliable sources, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

CRYPTO_4 `basic`

The messenger SHOULD NOT use implementations of cryptographic primitives and schemes that have been internally developed by the developer of the application.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of the implementation of cryptographic primitives and schemes used by the application;
2. Verify, for each implementation, that it is not internally developed by the controller;
3. Ask the controller, for each implementation, if the implementation of cryptographic primitives and schemes used by the application has been modified by the controller;
4. Repeat the previous steps for all possible frontends;
5. If one or more implementations is internally developed by the controller, and the auditor considers they should not be, ask the controller why;
6. If one or more implementations has been modified by the controller, and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no implementation of cryptographic primitives and schemes has been internally developped by the controller AND no implemenation of cryptographic primitives and schemes has been modified by the controller, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.8 Random numbers

RANDOM_1 `basic`

All random values SHOULD be generated using a secure cryptographic random number generator. [fSidI20, O.Random_1]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all random values, and how they are generated;
2. Verify, for each random value, if it is generated using a secure cryptographic random number generator. Such sources include, but are not limited to:
   - Android: `java.security.SecureRandom`
   - iOS: `SecRandomCopyBytes`
   - Linux: `/dev/random` and `/dev/urandom`
   - Windows: `CryptGenRandom`
3. Repeat the previous steps for all possible frontends;
4. If one or more random values are not generated using a secure cryptographic random number generator, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all random values are generated using a secure cryptographic random number generator, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

RANDOM_2 `basic`

The application MAY obtain random numbers from a source provided by the operating system.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all random values, and how they are generated;
2. Verify, for each random value, if it is generated using a source provided by the operating system. Such sources include, but are not limited to:
   - Android: `java.security.SecureRandom`
   - iOS: `SecRandomCopyBytes`
   - Linux: `/dev/random` and `/dev/urandom`
   - Windows: `CryptGenRandom`

3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, some random values are generated using a source provided by the operating system.

2.11.9 Data encryption & protection

FILE_1 `basic`

Files containing personal data or other sensitive data (e.g. keys) on the end device SHOULD be encrypted.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of files stored on the end device that can include personal or other sensitive data;
2. Ask the controller to provide, for each file containing personal or other sensitive data, if it is encrypted when at rest on the device;
3. Mark down if all files are encrypted when at rest on the device;
4. Repeat the previous steps for all possible frontends;
5. If one or more files containing personal data or other sensitive data are not encrypted, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all files containing personal data or other sensitive data are encrypted if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

FILE_2 `basic`

Files containing personal data or other sensitive data (e.g. keys) on the end device SHOULD NOT be stored in location accessible by other applications.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of files stored on the end device that can include personal data or other sensitive data;
2. Ask the controller to provide the storage location and the associated access control mechanism;

3. Verify that the location cannot be accessed by another application. Location that cannot be accessed by other application by default includes:
   - Android: the Internal Storage and not the External Storage.
   - iOS: the standard storage location
   - Web: session storage, local storage, indexDB ...
4. Repeat the previous steps for all possible frontends;;
5. If one or more files containing personal data or other sensitive data are stored in a location accessible by other application, and the auditor considers they should not be, ask the controller why;.

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all the files are stored in a location that cannot be accessed by other applications, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

FILE_3 `basic`

Personal data or other sensitive data SHOULD NOT be stored in files unless necessary.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of personal data or other sensitive data stored in files, as well as a justification for the necessity of storing the data in a file;
2. Verify, for each data stored in a file, that there is a valid justification for the necessity to store the data in a file;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, for all personal data or other sensitive data stored in a file, there is valid justification for the necessity to store the data in a file AND if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

FILE_4 `basic`

Personal data or other sensitive data (e.g. keys) stored in files MUST be deleted as soon as they are no longer needed.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of personal data or other sensitive data stored in files, along with their management process;
2. Verify, for each personal data or other sensitive data stored, that it is deleted as soon as it is no longer needed; Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all personal data or other sensitive data stored in files is deleted as soon as it is no longer needed. Otherwise, the requirement is not fulfilled (FAIL).

FILE_5 `basic`

Personal data and any other sensitive data SHOULD NOT be stored in cache files.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of cache files stored on the end device and their respective content;
2. Mark down if any of these cache files include personal data or any other sensitive data;
3. Repeat the previous steps for all possible frontends;
4. If any cache files contain personal data or any other sensitive data, and the auditor considers it should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, cache files do not contain personal or sensitive data, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

FILE_6 `basic`

File encryption methods used MUST be state of the art.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all encrypted files along with the encryption method;
2. Verify, for each encrypted files, that the encryption method is state of the art. Satisfactory encryption methods includes, but are not limited to:
   - Android: EncryptedFile and EncryptedSharedPreferences of the Security library[a]
   - iOS: file protection enabled (no `No Protection`) [b]

- Linux: `fscrypt`, `mcrypt`
- Windows: Encrypted File System[c]

3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the file encryption method is state of the art. Otherwise, the requirement is not fulfilled (FAIL).

[a] https://developer.android.com/topic/security/data
[b] https://developer.apple.com/documentation/uikit/protecting_the_user_s_privacy/encrypting_your_app_s_files
[c] https://learn.microsoft.com/en-us/windows/win32/fileio/file-encryption

2.11.10 Secure development & General coding / implementation recommendations

SEC_DEV_1 basic

All development support options (such as log calls, developer URLs, test methods, etc.) SHOULD be disabled in the production version. [fSidI20, O.Source_7]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all development support options (such as log calls, developer URLs, test methods, etc.) used during the development of the application;
2. Ask the controller, for each development support option, if the development process includes its disabling in the production version;
3. Mark down if, for each development support options, the development process includes its disabling in the production version;
4. Repeat the previous steps for all possible frontends;
5. If for one or more development support options, the development process does not include its disabling in the production version, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the development process includes disabling every develop support option for the production version, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SEC_DEV_2 basic

The controller MUST ensure that no debugging mechanisms remain in the production version. [fSidI20, O.Source_8]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of debugging mechanisms included in the development version;
2. Ask the controller to provide, for each used debugging mechanism, the measures taken to verify that it has not been included in the production version;
3. Verify that, for each used debugging mechanism, a measure to verify that it has not been included in the production version exists;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, a measure to verify that each used debugging mechanism has been removed in the production version, exists. Otherwise, the requirement is not fulfilled (FAIL).

2.11.10.1 WebViews / Javascript

This section applies only to Webviews

WEBVIEWS_1 `basic`

If the application switches to background mode, it SHOULD remove all sensitive data from the current view (Views in iOS and Activities in Android, respectively). [fSidI20, O.Plat_11]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of operations performed when the application is put in background;
2. Mark down if the list of operation includes the removal of all sensitive data from the current view;
3. Repeat the previous steps for all possible frontends;
4. If the application does not remove all sensitive data from the current view while put in the background, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the removal of sensitive data from the view is an operation performed when the application is put in background, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEBVIEWS_2 `basic`

The application SHOULD delete application-specific cookies after exiting. [fSidI20, O.Plat_13]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of operations performed when exiting the application;
2. Mark down if the list of operations includes the deletion of application-specific cookies;
3. Repeat the previous steps for all possible frontends;
4. If the list of operations does not include the deletion of application-specific cookies, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, application-specific cookies are deleted when exiting the application, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. . Otherwise, the requirement is not fulfilled (FAIL).

WEBVIEWS_3 `basic`

The application SHOULD implement URL whitelisting in the context of Webviews, in order to accept connections to URLs controlled or trusted by the controller.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of methods used to secure the call of URLs;
2. Mark down if the list of methods includes whitelisting in the context of WebViews;
3. Repeat the previous steps for all possible frontends;
4. If whitelisting in the context of WebViews is not used, and the auditor considers it should be, ask the controller why;;

Validation

If the application does not feature Webviews the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, whitelisting is used and enforced while accessing URLs in the context of WebViews, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEBVIEWS_4 basic

The application SHOULD disable local file and content access in the context of Webviews.

Prerequisite

- None;

Verification steps

1. Ask the controller if a Webview can open a local file or content;
2. Mark down if a Webview can open a local file or content;
3. Repeat the previous steps for all possible frontends;
4. If a Webview can open a local file or content and the auditor considers it should not, ask the controller why;

Validation

If the application does not feature Webviews the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, Webviews are not able to open local files or local content, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.10.2 Web front-end security The following requirements only apply to the Web frontend (if any).

WEB_SEC_1 basic

The Web application SHOULD be secured according to the state of the art.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide evidences demonstrating that the application is secured according to the state of the art;
2. Verify that the evidences provided by the controller demonstrate that the application is secured according to the state of the art;
3. If for one or more aspects, the application does not appear to be secured according to the state of the art, and the auditor considers it should be, ask the controller why;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if the auditor came to the conclusion that the application is secured according to the state of the art, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_2 `basic`

The Web application SHOULD use HTTP Strict Transport Security (HSTS). [ANS21b, R2]

Prerequisite

- None;

Verification steps

1. Ask the controller if the application uses HTTP Strict Transport Security (HSTS) for all HTTPS communications;
2. Mark down if the application uses HTTP Strict Transport Security (HSTS) for all HTTP communications;
3. If for one or more HTTPS communications, the application does not use HSTS, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if the application uses HTTP Strict Transport Security (HSTS) for all HTTP communications, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_3 `basic`

The Web application SHOULD NOT store personal data or sensitive data in local databases such as (localStorage, sessionStorage and IndexDB). [ANS21b, Sec. 5.5]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of the information stored by the application in local databases. Local databases include but are not limited to:
   - localStorage
   - sessionStorage
   - IndexDB
2. Verify that neither personal data nor sensitive data is stored in local databases;
3. If one or more personal data or sensitive data are stored in local databases, and the auditor considers they should not be, ask the controller why;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if no personal data nor sensitive data is stored in local databases, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_4 `basic`

Session cookies SHOULD NOT be accessible via JavaScript. [ANS21b, R30]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all session cookies used by the application;
2. Ask the controller, for each session cookie, if it has the `HttpOnly` attribute set;
3. Mark down if all session cookies have the `HttpOnly` attribute set;
4. If one or more session cookies do not have the `HttpOnly` attribute set, and the auditor considers they should, ask the controller why;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if all session cookies have the `HttpOnly` attribute set, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_5 `basic`

Cross-site transfer of Session Cookies SHOULD be disabled. [ANS21b, R33]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all session cookies used by the application;
2. Ask the controller, for each session cookie, if it has the `Same site` attribute set to `Strict`;
3. Mark down if all session cookies have the `Same site` attribute set to `Strict`;
4. If one or more session cookies do not have the `Same site` attribute set to `Strict`, and the auditor considers they should, ask the controller why;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if all session cookies have the `Same site` attribute set to `Strict`, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_6 `basic`

Session cookies SHOULD NOT be sent over unsecured communication channels. [ANS21b, R31]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all session cookies used by the application;
2. Ask the controller, for each session cookie, if it has the `Secure` attribute set;
3. Mark down if all session cookies have the `Secure` attribute set;
4. If one or more session cookies do not have the `Secure` attribute set, and the auditor considers they should, ask the controller why;

1. `Secure` attribute set;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if all session cookies have the `Secure` attribute set, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_7 `basic`

Cookies MUST only be set once they are needed.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all cookies used by the application, along with the steps in which they used are and when they are set;
2. Verify, for each cookie, that it is set just before the first step in which it is needed;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). . Otherwise, the requirement is fulfilled (PASS), all cookies are set just before the first step in which they are neededOtherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_8 `basic`

Cookies MUST be deleted as soon as they are no longer needed.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all cookies used by the application, along with the steps

in which they are used and when they are deleted;

2. Verify, for each cookie, that it is deleted just after the last step in which it is needed;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if all cookies are deleted just after the last step in which they are needed, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_9 `basic`

Session identifier SHOULD not be included in the URL. [Fou21, A07_2021]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of session identifiers;
2. Ask the controller, for each session identifier, if it can be included in a URL;
3. Mark done if any session identifier can be included in an URL;
4. If one or more session identifier can be included in an URL, and the auditor considers it should not, ask the controller why;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if no session identifier is included in an URL, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_10 `basic`

Session identifier SHOULD be invalidated after logout. [Fou21, A07_2021]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all session identifiers and when they are invalidated by the application;
2. Mark down all session identifiers that are invalidated by the application after the user logs out;
3. If one or more session identifiers are not invalidated by the application after the user logs out, and the auditor considers they should be, ask the controller why;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if all session identifiers are invalidated after the user logs out, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_11 `basic`

Session identifier SHOULD be invalidated after idle timeout. [Fou21, A07_2021]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all session identifiers and when they are invalidated by the application;
2. Mark down all session identifiers that are invalidated by the application after an idle timeout;
3. If one or more session identifiers are not invalidated by the application after an idle timeout, and the auditor considers they should be, ask the controller why;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if all session identifiers are invalidated after an absolute timeout, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

WEB_SEC_12 `basic`

Session identifier SHOULD be invalidated after absolute timeouts. [Fou21, A07_2021]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all session identifiers and when they are invalidated by the application;
2. Mark down all session identifiers that are invalidated by the application after an absolute timeout;
3. If one or more session identifiers are not invalidated by the application after an absolute timeout, and the auditor considers they should be, ask the controller why;

Validation

If the messenger does not feature a Web frontend the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS), if all session identifiers are invalidated after an absolute timeout, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.11 Logs

LOGS_1 `basic`

The application SHOULD not write any personal data in the logs.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all possible logfiles and their content;
2. Mark down if any of the logfiles contains any personal data;
3. Repeat the previous steps for all possible frontends;
4. If one or more logfiles contain personal data, and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, logfiles do not include personal data, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

LOGS_2 `basic`

The application SHOULD not write any keys in the logs.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of all possible logfiles and their content;
2. Mark down if any of the logfiles contains any keys;
3. Repeat the previous steps for all possible frontends;
4. If one or more logfiles contain keys, and the auditor considers they should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, logfiles do not include keys, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

LOGS_3 `basic`

The logs SHOULD be encrypted at rest.

Prerequisite

- None;

Verification steps

1. Ask the controller if the logs are encrypted at rest;
2. Mark down if the logs are encrypted at rest;
3. Repeat the previous steps for all possible frontends;
4. If one or more logs are not encrypted at rest, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the logs are encrypted at rest, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.12 Third party software

THRDP_SOFT_1 `basic`

Third-party libraries and frameworks SHOULD be used in their latest available version for the platform operating system in use. [fSidI20, O.TrdP_1]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of third-party libraries and frameworks used by the application, along with their version;
2. For each library or framework consult the official distribution source and inspect the latest version available for the operating system used by the current frontend;
3. Mark down if for each library or framework the version used is the latest available;
4. Repeat the previous steps for all possible frontends;
5. If for one or more library the version used are not the latest available, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all third-party libraries and frameworks used is the latest available version, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

THRDP_SOFT_2 `basic`

The controller MUST perform regular checks of third-party libraries and frameworks with regard to vulnerabilities. [fSidI20, O.TrdP_2]

Prerequisite

- None;

Verification steps

1. Ask the controller if there exists a process to regularly check if third-party libraries and frameworks include vulnerabilities;
2. Mark down if a process to regularly check vulnerabilities in third-party libraries and frameworks exists;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, a process to regularly check vulnerabilities in third-party libraries and frameworks exists. Otherwise, the requirement is not fulfilled (FAIL).

THRDP_SOFT_2.a `basic`

Functions from libraries and frameworks MUST NOT be used if any vulnerabilities are known. [fSidI20, O.TrdP_2]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the policy with regard to identified vulnerabilities in third-party libraries and frameworks;
2. Mark down if the policy includes not to use functions with known vulnerabilities;
3. Mark down if the policy includes appropiate steps to patch or remove the affected libraries and frameworks in a timely manner;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, a policy specify not to use functions with known vulnerabilities AND appropiate steps to patch or remove the affected functions in a timely manner exist. Otherwise, the requirement is not fulfilled (FAIL).

THRDP_SOFT_3 `basic`

The application SHOULD not disclose personal data to third-party libraries. [fSidI20, O.TrdP_6]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of third-party libraries and frameworks used by the application;
2. Ask the controller, for each third-party library or framework used, to provide a list of all data disclosed to it;
3. Verify that the data disclosed to third-party libraries and frameworks does not includes personal data; Repeat the previous steps for all possible frontends;
4. If one or more personal data are disclosed to third-party libraries or frameworks, and the auditor considers they should not be, ask the controller why;.

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the data disclosed to third-party libraries and frameworks does not include personal data, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

THRDP_SOFT_4 `basic`

Before using third-party libraries and frameworks, their source SHOULD be checked for trustworthiness. [fSidI20, O.TrdP_5]

Prerequisite

- None;

Verification steps

1. Ask the controller to describe the process used to obtain the software associated to third-party libraries and frameworks;
2. Verify that the source of the software is checked for trustworthiness before using it. Checking trustworthiness can be done by, but is not limited to, reviewing the following elements:
   - reputation of the vendor
   - number of users/download
   - level of activity of the development/maintenance team
3. Repeat the previous steps for all possible frontends;
4. If the controller does not check the trusworthiness of third-party libraries or frameworks before using them, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller checks the trustworthiness of third-party libraries and frameworks source before using it, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

THRDP_SOFT_5 `basic`

Data received via third-party libraries and frameworks SHOULD be validated. [fSidI20, O.TrdP_7]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of third-party libraries and frameworks used by the application;
2. Ask the controller to describe how data received from third-party libraries is validated.
3. Verify that data received from third-party libraries is validated. Data validation includes, but it not limited to:
   - data type validation
   - range and constraint validation
   - code and cross-reference validation
   - structured validation
   - consistency validation
4. Repeat the previous steps for all possible frontends;
5. If one or more data received throug third-party libraries is not validated, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all data received via third-party libraries are validated, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

THRDP_SOFT_6 `basic`

Third-party software that is no longer kept up-to-date by the manufacturer or developer SHOULD NOT be used. [fSidI20, O.TrdP_8]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of third-party software used by the application;
2. For each library or framework consult the official distribution source and mark down the release date of the latest update;
3. Repeat the previous steps for all possible frontends;
4. If for one or more third-party software used, the latest release is older than 12 months, and the auditor considers it should not be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the latest release of all third party software used is not older than 12 months, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.13 Software updates

UPDATES_1 `basic`

The messenger application SHOULD feature a mechanism to allow updates.

Prerequisite

- None;

Verification steps

1. Ask the controller if the application features a mechanism to allow updates;
2. Mark down if the controller declared that the application features a mechanisms to allow updates;
3. Repeat the previous steps for all possible frontends;
4. If the application does not feature a mechanism to allow updates, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller declared that the application features a mechanism to allow updates, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

UPDATES_2 `basic`

The messenger application SHOULD inform the user when an update is available.

Prerequisite

- None;

Verification steps

1. Ask the controller how the application informs the user when an update is available;
2. Mark down if the process to inform the user of new updates includes notifications within the application;
3. Repeat the previous steps for all possible frontends;
4. If the application does not inform the user within the application when an update is available, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller declared that the application informs the user when an update is available within the application, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

UPDATES_3 `basic`

The messenger application SHOULD verify the integrity of software updates before installation.

Prerequisite

- None;

Verification steps

1. Ask the controller for details on the update process used by the application;
2. Mark down if the update process includes integrity checks by the application prior to executing the update;
3. Mark down the reaction if the integrity check fails;
4. Repeat the previous steps for all possible frontends;
5. If the application does not verify the integrity of software updates before installation, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application performs an integrity check prior to executing the update AND aborts applying the update if the check fails, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

UPDATES_4 `basic`

The messenger application SHOULD verify authenticity of software updates before installation.

Prerequisite

- None;

Verification steps

1. Ask the controller for details on the update process used by the application;

2. Mark down if the update process includes authenticity checks by the application prior to executing the update;
3. Mark down the reaction if the authenticity check fails;
4. Repeat the previous steps for all possible frontends;
5. If the application does not verify the authenticity of software updates before installation, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application performs an authenticity check prior to executing the update AND aborts applying the update if the check fails, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

UPDATES_5 `basic`

For each update related to security, an application that has not been updated SHOULD refuse to work after a grace period. The duration of the grace period SHOULD be based on the criticality of the vulnerability corrected by the update.

Prerequisite

- None;

Verification steps

1. Ask the controller how long an outdated application can continue to work after a security-related update has been released;
2. Mark down if, after a security-related update has been released, an outdated application will stop working after a grace period;
3. Ask the controller how the duration of the grace period is selected;
4. Mark down if the duration of the grace period is based on the criticality of the vulnerability corrected by the update;
5. Repeat the previous steps for all possible frontends;
6. If after a security-related update has been released, an outdated application will not stop working after a grace period, and the auditor considers it should, ask the controller why;
7. If the duration of the grace period is not based on the criticality of the vulnerability corrected by the update, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, an outdated application will stop working after a grace period AND the duration of the grace period is based on the criticality of the vulnerability corrected by the update, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.14 Software distribution

DISTRIB_1 `basic`

The controller SHOULD offer to the user a way to verify the authenticity and integrity of the application and updates before installation.

Prerequisite

- None;

Verification steps

1. Ask the controller on how users are able to verify the authenticity and integrity of the application and updates before installation;
2. Mark down if the user is able to perform authenticity and integrity checks. In order to evaluate this, the auditor may include the following markers:
   - Availability of digital signatures
   - Availability of hash values
3. Repeat the previous steps for all possible frontends;
4. If the user does not have an option to verify the authenticity or integrity of the application or updates, and the auditor considers they should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the user has a way to verify the authenticity and integrity of the application and updates, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

DISTRIB_2 `basic`

The controller SHOULD offer a way to directly obtain the application from the controller.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide a list of distribution channels;
2. Verify that the list includes a commonly used channel to directly obtain the application from the controller;
3. Repeat the previous steps for all possible frontends;
4. If there is not a commonly used channel to directly obtain the application from the controller, and the auditor considers it should be, ask the controller why;.

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the controller provides a way to directly obtain the application from the controller, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

DISTRIB_3 `basic`

The controller MAY distribute the messenger through third party platforms in addition to direct distribution channels from the controller.

Prerequisite

- None;

Verification steps

1. Visit the application distribution platform of the current operating system, and search for the messenger application;
2. Mark down if the application is available on the application distribution platform;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application is available on the application distribution platform. Otherwise, the requirement is not fulfilled (FAIL).

2.11.15 Authentication

AUTH_1 `basic`

The messenger system SHOULD enforce a password policy according to the state of the art.

Prerequisite

- None;

Verification steps

1. Consult the password policy by going through the account creation process;
2. Verify that the password policy is compliant with the state of the art. The auditor can consult national guidelines or resources such as the following to assess this compliance:
   - CNIL's Deliberation no. 2017-012 of 19 January 2017 on the adoption of a recommendation relating to passwords [CNI17]
   - BSI's Recommendations on Creating Secure Passwords [fIS]
3. If the password policy is not compliant with the state of the art, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if the password policy is compliant with the state of the art, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

AUTH_2 `basic`

The messenger SHOULD provide a way to change the password and other authentication tokens.

Prerequisite

- An account A.

Verification steps

1. Login with the account A;
2. Search for the option to change the password;
3. Mark down if the option to change the password exists;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the option to change the password exists, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

AUTH_3 `basic`

The messenger MAY feature a second authentication factor.

Prerequisite

- An account A;

Verification steps

1. Login with the account A;
2. Search for the option to use a second authentication factor;
3. Mark down if the option to use a second authentication factor exists;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, if the option to use a second authentication factor exists. Otherwise, the requirement is not fulfilled (FAIL).

Subsequent steps

If this requirement is not satisfied (FAIL), the following requirements are not applicable (NA): AUTH_4

AUTH_4 `basic`

The messenger SHOULD use open standards for a second authentication factor.

Prerequisite

- An account A;

Verification steps

1. Login with the account A;
2. Consult the list of second factor authentication methods supported by the application;
3. Verify if the supported second factor authentication methods use open standards. Open standards for second factor authentication methods include, but are not limited to:
   - U2F, Universal 2nd Factor
   - TOTP, Time-Based One-Time Password
   - HOTP, Hash-Based One-Time Password
4. Repeat the previous steps for all possible frontends;
5. If the messenger does not support a second factor authentication method following an open standard, and the auditor considers it should, ask the controller why;

Validation

If the messenger does not feature a second authentication factor the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, the messenger supports a second factor authentication method following an open standard, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

AUTH_5 `basic`

The messenger MUST NOT use SMS for a second authentication factor.

Prerequisite

- An account A;

Verification steps

1. Login with the account A;
2. Consult the list of second factor authentication supported by the application;
3. Mark down if the application supports SMS as a second authentication factor;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application does not support SMS as a second authentication factor. Otherwise, the requirement is not fulfilled (FAIL).

AUTH_6 `basic`

For authentication based on a username and a password, the strength of the password used MAY be displayed to the user. Information regarding the strength of the chosen password MUST NOT be retained in the application memory or backend. [fSidI20, O.Auth_8].

Prerequisite

- None;

Verification steps

1. Proceed to the account creation procedure until reaching the password configuration step;
2. Mark down if the interface features an indicator for the strength of the password;
3. Ask the controller if information about the strength of the password is stored or transmitted outside of the interface;
4. Mark down if information about the strength of the password is stored or transmitted outside of the interface.

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the interface features an indicator for the strength of the password AND no information about the strength of the password is stored nor transmitted outside of the interface. Otherwise, the requirement is not fulfilled (FAIL).

AUTH_7 `basic`

When credentials are entered via the keyboard, the application SHOULD prevent recordings from becoming visible to third parties. This specifically excludes auto-correction and auto-complete functions, third-party input keyboards and any form of storage that can be evaluated by third parties. [fSidI20, O.Data_9]

Prerequisite

- None;

Verification steps

1. Proceed to the account creation procedure until reaching the password configuration step;
2. Mark down if the field receiving the password conceals the user input;
3. Start entering a common name in the password field (e.g. ”keybo”);
4. Mark down if the keyboard suggests a completion;
5. Enter a common name with a spelling error (e.g ”keuboard”);
6. Mark down if the keyboard suggests a correction;
7. Repeat the previous steps for all possible frontends;
8. If the field receiving the password does not conceal the user input, and the auditor considers it should, ask the controller why;

9. If the keyboard suggests a completion, and the auditor considers it should not, ask the controller why;
10. If the keyboard suggests a correction, and the auditor considers it should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the field receiving the password conceals the user input AND no completion was suggested AND no correction was suggested, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

AUTH_8 `basic`

The messenger SHOULD require a second level authentication before accessing sensitive data or modifying sensitive settings.

Prerequisite

- An account A.

Verification steps

1. Login with account A;
2. Navigate to the menu to change a sensitive setting;
3. Mark down if a second level of authentication was requested to allow changing sensitive setting;
4. Repeat the previous steps for all possible frontends;
5. If a second level of authentication was not requested, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, a second level of authentication was requested, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.15.1 Lock / logout

LOCK_1 `basic`

The messenger application SHOULD provide a way to logout the user from the messenger service.

Prerequisite

- An account A.

Verification steps

1. Login with the account A;
2. Locate the option to logout;

3. Proceed to log out;
4. Restart the application;
5. Mark down if the application is still logged in with account A;
6. Repeat the previous steps for all possible frontends;
7. If the application does not offer a way to logout the user, and the auditor considers it should have been, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the user has a way to log out AND the user is logged in after proceeding with the logout, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

LOCK_2 `basic`

The messenger application SHOULD provide a way to lock the application.

Prerequisite

- A populated account A.

Verification steps

1. Search for the option to lock the application;
2. Mark down if the option to lock the application exists;
3. Enable the application lock feature;
4. Close the application;
5. Open the application;
6. Mark down if the user is prompted to unlock the application;
7. Repeat the previous steps for all possible frontends;
8. If the option to lock the application does not exist, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the option to lock the application exists AND if the conversation is not accessible, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.15.2 Re-authentication

REAUTH_1 `basic`

The messenger MAY feature a re-authentication mechanism such as a PIN code, a passphrase, or other authentication mechanisms.

Prerequisite

- None;

Verification steps

1. Ask the controller if the application features a re-authentication mechanism such as a PIN code, a passphrase, or other authentication mechanisms.
2. Mark down if the application features a re-authentication mechanism;
3. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application features a re-authentication mechanism. Otherwise, the requirement is not fulfilled (FAIL).

REAUTH_2 `basic`

The application MAY require re-authentication when it is launched or after a period of inactivity. [fSidI20, O.Auth_11]

Prerequisite

- An account A.

Verification steps

1. Ask the controller is re-authentication is required when the application is launched or after a period of inactivity.
2. Mark down if re-authentication is required when the application is launched;
3. Mark down the time period of inactivity after which a re-authentication is requested by the application;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, re-authentication is required when the application is launched or after a period of inactivity. Otherwise, the requirement is not fulfilled (FAIL).

REAUTH_3 `basic`

The messenger MAY require re-authentication before accessing sensitive data or modifying sensitive settings.

Prerequisite

- An account A.

Verification steps

1. Login with account A;
2. Navigate to the menu to change the password and change it for a new value;
3. Mark down if re-authentication was required to allow changing the password;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, re-authentication was required. Otherwise, the requirement is not fulfilled (FAIL).

2.11.16 Stateful and stateless authentication

2.11.16.1 Stateful authentication

SF_AUTH_1 `basic`

Session handling SHOULD be implemented using secure frameworks. [fSidI20, O.Sess_1]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of frameworks used for session handling by the application;
2. Verify that all frameworks used for sessions handling are considered a secure framework;
3. Repeat the previous steps for all possible frontends;
4. If one or more frameworks used for session handling is not considered to be secure, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all frameworks used for session handling are secure frameworks, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SF_AUTH_2 `basic`

Session identifiers SHOULD be protected as sensitive data. [fSidI20, O.Sess_3]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of all processed data that is considered to be sensitive and the respective protection mechanisms;
2. Mark down if session identifiers are listed;
3. Mark down if session identifiers are protected in similar means as other sensitive data;
4. Repeat the previous steps for all possible frontends;
5. If session identifiers are not protected by similiar means as other sensitive data, and the auditor considers they should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all session identifiers are protected by similar means as other sensitive data, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SF_AUTH_3 `basic`

Session identifiers MUST NOT be stored unencrypted on permanent storage media. [fSidI20, O.Sess_4]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of session identifiers;
2. Ask the controller, for each session identifier, if it is stored on a permanent storage media;
3. Ask the controller, for each session identifier stored on a permanent storage media, if it is stored encrypted;
4. Markdown if all sessions identifiers stored on a permanent storage media are stored encrypted;
5. Repeat the previous steps for all possible frontends;
6. If one or more session identifier stored on a permanent storage media are stored unencrypted, and the auditor considers it should not be, ask the controller why;.

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all session identifiers stored on a permanent storage media are stored encrypted, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SF_AUTH_4 `basic`

The application and its backend SHOULD actively terminate the application session after an appropriate session timeout, according to current best practice recommendations. [fSidI20, O.Sess_5]

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of sessions handled by the application and the corresponding timeouts;
2. Markdown if all sessions feature a timeout;
3. Ask the controller, for each session, what happen when the timeout is reached;
4. Verify that, for all sessions, it is invalidated upon reaching the timeout;
5. Repeat the previous steps for all possible frontends;
6. If one or more sessions do not feature a timeout, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all sessions feature a timeout AND all sessions are invalidated upon reaching the timeout, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SF_AUTH_5 `basic`

When an application session is terminated, the application SHOULD securely delete the session identifier in the device.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of sessions handled by the application;
2. Ask the controller, for each session, what happen to the corresponding session identifier when the session is terminated;
3. Mark down if all session identifiers are deleted;
4. Repeat the previous steps for all possible frontends;
5. If for one or more sessions identifiers are not deleted after terminating the session, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all sessions identifiers are deleted once the corresponding session is terminated, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.16.2 Stateless authentication

SL_AUTH_1 `basic`

The authentication token SHOULD be kept in a secure memory area on the device (e.g. KeyChain/KeyStore).

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of authentication tokens used by the application along with its storage location on the device;
2. Verify, for each authentication token, that it is stored in a secure memory area. Secure memory areas include, but are not limited to:
   - Android: `KeyStore`, `KeyChain`
   - iOS: `KeyStore`, `Secure Enclave`
3. Repeat the previous steps for all possible frontends;
4. If one or more authentication tokens are not kept in a secure memory area, and the auditor considers they should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all authentication token are stored in a secure memory area, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SL_AUTH_2 `basic`

Sensitive data MUST NOT be embedded in an authentication token. [fSidI20, O.Tokn_2].

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of authentication tokens used by the application along with all data embedded in the token;
2. Verify, for each authentication token, that it does not embed sensitive data;
3. Repeat the previous steps for all possible frontends;
4. If one or more authentication tokens embed sensitive data, and the auditor considers they should not, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, no authentication token embeds sensitive data, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

SL_AUTH_3 `basic`

The private key used to sign the authentication token MUST NOT be present in the device. [fSidI20, O.Tokn_5].

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the list of authentication tokens used by the application;
2. Ask the controller to provide, for each authentication token, the identity of the private key used to signed it along with its location;
3. Verify, for each authentication token, that the private key used to sign the authentication token is not present in the device;
4. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, all private keys used to sign the authentication token are not located on the device.Otherwise, the requirement is not fulfilled (FAIL).

SL_AUTH_4 `basic`

The messenger application SHOULD feature an option to invalidate all previously issued authentication tokens (for instance, if the device was lost).

Prerequisite

- A configured account A connected on a device d_1;

Verification steps

1. Search for the option to invalidate all previously issued authentication tokens for the account;
2. Mark down if the option to invalidate all previously issued authentication tokens for the account exists;
3. Close the application on device d_1;
4. Use the option to invalidate all previously issued authentication tokens for account A;
5. Start the application on device d_1;
6. Mark down if the user is not authenticated with the account A on device d_1;
7. Repeat the previous steps for all possible frontends;
8. If the option to invalidate all previously issued authentication tokens for the account does not exist, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the option to invalidate all previously issued authentication tokens for the account exists AND the user is not authenticated with the account A on device d_1, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.17 Resilience

RESI_1 `basic`

The messenger application MAY check if the device is rooted / jailbroken and inform the user of the associated risks. [fSidI20, O.Resi_2]

Prerequisite

- None;

Verification steps

1. Ask the controller if the application checks if the device is is rooted / jailbroken;
2. Mark down if the application checks if the device is rooted / jailbroken;
3. Ask the controller if the application informs the user of the associated risks in case the device is rooted / jailbroken;
4. Mark down if the application informs the user of the associated risks;
5. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application checks if the device is rooted / jailbroken AND informs the user of the associated risks. Otherwise, the requirement is not fulfilled (FAIL).

RESI_2 `basic`

The application SHOULD reliably detect and prevent the start in a development/debug environment. [fSidI20, O.Resi_3]

Prerequisite

- None;

Verification steps

1. Ask the controller if the application detects if it is running in a development/debug environment and the corresponding application response;
2. Mark down if the application prevents its start if it detected that it is running in a development/debug environment;
3. Repeat the previous steps for all possible frontends;

4. If the application does not check if it is running in a development/debug environment, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application detects if it is running in a development/debug environment AND prevents running in a development/debug environment, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.18 Backup & recovery

BACKUP_1 `basic`

Backups SHOULD be encrypted.

Prerequisite

- None;

Verification steps

1. Ask the controller if the backups are encrypted;
2. Mark down if the backups are encrypted;
3. Repeat the previous steps for all possible frontends;
4. If backups are not encrypted and and the auditor considers they should be, ask the controller why;

Validation

If the application does not support backups the requirement is not applicable (NA). Otherwise, the requirement is fulfilled (PASS) if, for all available frontends, The requirement is fulfilled (PASS) if, for all available frontends, backups are encrypted, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

BACKUP_2 `basic`

The application SHOULD provide a way to create a backup of the data stored locally.

Prerequisite

- A populated account

Verification steps

1. Search for the process to create a backup;
2. Proceed to generate a backup;
3. Mark down if during the backup process an option was available to store the backup locally;

4. Select to store the backup locally;
5. Mark down if the backup was storad locally;
6. Repeat the previous steps for all possible frontends;
7. If the backup can not be stored locally, and the auditor considers they should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application provides a way to store a local backup, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

BACKUP_3 `basic`

The application MAY offer cloud backups in addition to local backups.

Prerequisite

- None;

Verification steps

1. Ask the controller if the application offers local backups;
2. Mark down if the application offers local backups;
3. Ask the controller if the application offers cloud backups;
4. Mark down if the application offers cloud backups;
5. Repeat the previous steps for all possible frontends;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application offers local backups AND the application offers cloud backups. Otherwise, the requirement is not fulfilled (FAIL).

BACKUP_4 `basic`

The application SHOULD provide a way to restore the data from a previous backup.

Prerequisite

- A populated account A.

Verification steps

1. Login with the account A, and proceed to the backup procedure;
2. Uninstall the application, then re-install it;
3. Login with the account A, and proceed to the backup restoration procedure;
4. Mark down if the backup restoration successfully finished;

5. Inspect the application settings and the content of the discussions;
6. Mark down if all the settings and content of the discussion is correctly restored;
7. Repeat the previous steps for all possible frontends;
8. If one or more elements of the settings or discussion is not restored, and the auditor considers it should be, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the application provides an option to restore the content from a backup AND the restoration process was successful, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.19 Account recovery

ACC_REC_1 `basic`

The messenger SHOULD provide a mean to recover an account in case of a credential loss.

Prerequisite

- an account A;

Verification steps

1. Logout with account A;
2. Search for the option to recover the account in case of a credential loss;
3. Mark down of the option to recover the account in case of a credential loss exists;
4. Proceed to the account recovery procedure without using the original credential;
5. Mark down if the account recovery procedure finished successfully: the auditor recovered a new credential and successfully logged in with the account;
6. Repeat the previous steps for all possible frontends;;
7. If the option to recover the account in case of a credential loss does not exist, and the auditor considers it should, ask the controller why;

Validation

The requirement is fulfilled (PASS) if, for all available frontends, the option to recover the account in case of a credential loss exists AND the account recovery procedure finished successfully, OR if the auditor is satisfied with the requested explanation given by the controller or decided against requesting an explanation. Otherwise, the requirement is not fulfilled (FAIL).

2.11.20 Security certification and adherence to code of conduct

SEC_CERTIF_1 `basic`

The controller MAY demonstrate compliance with the GDPR for some or all of these requirements by providing a certificate if it adheres to one or more appropriate and approved certification mechanism as marked down in Art. 42 GDPR.

Prerequisite

- None;

Verification steps

1. Ask the controller to provide the Art. 42 GDPR certificate(s) alongside a list of every requirement that it wishes to show compliance with the provided Art. 42 GDPR certificate(s) along with the corresponding scope and verification mechanisms of the Art. 42 GDPR certificates(s);
2. For every requirement that the controller wishes to show compliance with perform the following steps:
   (a) Check every requirement of all the provided certificates for a match with the corresponding requirement from this cataloge;
   (b) Check that the corresponding verifciation method of the provided Art. 42 GDPR certificate is sufficient to achieve the same level of verification as the corresponding verification method of this cataloge;
   (c) Check that the corresponding requirement of the provided Art. 42 GDPR certificate was fulfilled (PASS);

Validation

For every requirement that passes the three checks, the corresponding recquirement of this cataloce is to be considered fulfilled (PASS).

A Procedures

This section details a number of procedure whose execution is required before or during the verification of some requirements.

A.1 Application usage procedure

A.1.1 Create an account

1. Go through the standard registration procedure to create an account A.

A.1.2 Create set of test accounts

1. Using the create account procedure (A.1.1), create 5 accounts A_1 to A_5.
2. Using account A_1 create a group G_1; join the group G_1 with accounts A_2 to A_5.
3. Using account A_1 create a group G_2; join the group G_2 with accounts A_2.

A.1.3 Login with an account

- Start the application / open the website and connect with the account A.

A.1.4 Populate an account

1. Login with the account A (A.1.3)
2. Add contacts
   (a) If the set of test account $\mathcal{A}$ does not exist, create it (A.1.2)
   (b) Add as contact the first five accounts of the set of test account $\mathcal{A}$
3. Join group G_1 with account A
4. Populate conversations
   (a) Using account A, send one message , one picture and one URL to each contact;
   (b) Using each account of $\mathcal{A}$, send one message, one picture and one URL to A;
   (c) Using each account of $\mathcal{A}$, send one message , one picture and one URL to each group that includes A;

A.1.5 Standard use procedure

- Launch the application or load the service web page, and login with the account A;
- For each available view or panel open all the menus;
- For each contact / group, visit all corresponding sub-menus;
- For each contact / group, open the last media (picture, video, animated GIF)
- For each contact / group, send the following:
  - A text message
  - A picture
  - A video
  - An animated GI
  - An emoji
  - A URL
- Put the application in background for 5 minutes and then put it back in foreground;
- Repeat steps 5 to 6;
- Leave the device idle until the screen turns off, then wait 5 minutes and open the application again;
- Repeat steps 5 to 6;
- Log out with the account A

A.1.6 Writing and sending a message

When instructed to send a message, randomly select a sentence in the GDPR that contains at least 10 words.

A.1.7 Identify hosts

- Start a network traffic capture;
- Inspect the traffic capture with Wireshark;
- Select DNS queries using the filter `dns && dns.flags.response == 0`;
- Get the list of of all names in the DNS queries (field `dns.qry.name`);

B Tools

B.1 General environment

- Computer or VM with KALI Linux installed (https://www.kali.org/)

B.2 Network tools

B.2.1 Passive network traffic capture environment

- Wi-Fi Access point[1]
- Wireshark

B.2.2 Encrypted network traffic capture environment (MITM)

- mitmproxy https://mitmproxy.org/

References

[ANS17] ANSSI. Security Recommendations for TLS, January 2017.

[ANS21a] ANSSI. Guide de sélection d'algorithmes cryptographiques. Technical report, August 2021.

[ANS21b] ANSSI. Recommandations pour la mise en œuvre d'un site web : maîtriser les standards de sécurité côté navigateur. Technical report, April 2021.

[CNI17] CNIL. Deliberation no. 2017-012 of 19 January 2017 on the adoption of a recommendation relating to passwords. Technical report, January 2017.

[CNI18] CNIL. Analyse d'impact relative à la protection des données (AIPD) 3 : les bases de connaissances. Technical report, 2018.

[EDP22] EDPB. Guidelines 01/2022 on data subject rights - Right of access. Technical report, January 2022.

[EDP23] EDPB. Guidelines 03/2022 on Deceptive design patterns in social media platform interfaces: how to recognise and avoid them. Technical Report Version 2.0, February 2023.

[ENI14a] ENISA. Algorithms, Key Sizes and Parameters Report. Technical report, 2014.

[ENI14b] ENISA. *Study on cryptographic protocols.* Publications Office, November 2014.

[ENI17] ENISA. *Handbook on security of personal data processing.* Publications Office, LU, 2017.

[1] https://cybergibbons.com/security-2/quick-and-easy-fake-wifi-access-point-in-kali/

[fIS] Federal Office for Information Security. Creating Secure Passwords.

[fIS23] Federal Office for Information Security. BSI TR-02102-1: Cryptographic Mechanisms: Recommendations and Key Lengths. Technical report, January 2023.

[fIS24] Federal Office for Information Security. Standardised messenger audit d1 - frontend requirements. Technical report, 2024.

[Fou21] The OWASP Foundation. OWASP Top 10: 2021, 2021.

[fSidI20] Bundesamt für Sicherheit in der Informationstechnik. Security requirements for eHealth applications. Technical report, 2020.

[GSBM23] Colin M. Gray, Cristiana Santos, Nataliia Bielova, and Thomas Mildner. An Ontology of Dark Patterns Knowledge: Foundations, Definitions, and a Pathway for Shared Knowledge-Building, September 2023. arXiv:2309.09640 [cs].

[IAN23] IANA. Transport Layer Security (TLS) Parameters, September 2023.

[NIS19] NIST. Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations. Technical Report NIST Special Publication (SP) 800-52 Rev. 2, National Institute of Standards and Technology, August 2019.

[WP214a] WP29. Opinion 05/2014 on Anonymisation Techniques. Technical report, October 2014.

[WP214b] WP29. Opinion 9/2014 on the application of Directive 2002/58/EC to device fingerprinting. Technical report, November 2014.

[WP217] WP29. Guidelines on the right to data portability. Technical report, May 2017.


edpb
European Data Protection Board