



## Assignment of bachelor's thesis

<b>Title:</b>	Analysis of the CTU Teaching Survey's Anonymity
<b>Student:</b>	Eliška Helikarová
<b>Supervisor:</b>	Ing. Josef Kokeš, Ph.D.
<b>Study program:</b>	Informatics
<b>Branch / specialization:</b>	Computer Security and Information technology
<b>Department:</b>	Department of Computer Systems
<b>Validity:</b>	until the end of summer semester 2023/2024

### Instructions

- 1) Describe the CTU Teaching Survey application: It's purpose, history, features, publicly available technical information.
- 2) Acquire a developer access to the application. Study the application's source code and data structure.
- 3) Perform a threat analysis of the application. List potential attackers and their motivations, propose possible attack vectors, evaluate the impact of discovered threats.
- 4) Analyze the application, targeting the previously identified threats, with a particular focus on those that impact the student's anonymity.
- 5) Evaluate your findings. Discuss available methods of breaching the student's anonymity. Propose countermeasures on the side of the application, as well as on the side of the student.



Bachelor's thesis

# **ANALYSIS OF THE CTU TEACHING SURVEY'S ANONYMITY**

**Eliška Helikarová**

Faculty of Information Technology  
Department of Computer Systems  
Supervisor: Ing. Josef Kokeš, Ph.D.  
May 11, 2023

Czech Technical University in Prague  
Faculty of Information Technology

© 2023 Eliška Helikarová. All rights reserved.

*This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).*

Citation of this thesis: Helikarová Eliška. *Analysis of the CTU Teaching Survey's Anonymity*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

# Contents

<b>Acknowledgments</b>	<b>vi</b>
<b>Declaration</b>	<b>vii</b>
<b>Abstract</b>	<b>viii</b>
<b>Abbreviations</b>	<b>x</b>
<b>Introduction</b>	<b>1</b>
<b>1 The CTU Survey Application Overview</b>	<b>3</b>
1.1 Purpose of the CTU Survey . . . . .	3
1.2 Survey modules . . . . .	3
1.3 System roles . . . . .	4
1.4 Lifecycle of a survey instance . . . . .	4
1.5 History . . . . .	5
1.6 Technologies used . . . . .	6
1.7 Infrastructure . . . . .	6
<b>2 Anonymity</b>	<b>9</b>
2.1 Definition of anonymity . . . . .	9
2.2 Privacy vocabulary . . . . .	10
2.3 Privacy and security . . . . .	10
2.4 Importance of anonymity in Anketa CTU . . . . .	11
<b>3 Threat Analysis</b>	<b>13</b>
3.1 Threat modeling . . . . .	13
3.1.1 Decomposing the application . . . . .	13
3.1.2 Determining and ranking threats . . . . .	14
3.1.3 Countermeasures and mitigations . . . . .	15
3.2 LINDDUN framework . . . . .	15
3.2.1 Privacy threat categories . . . . .	15
3.2.2 Methodology . . . . .	17
<b>4 Threat Analysis of the CTU Survey</b>	<b>19</b>
4.1 Threat model . . . . .	19
4.2 Mapping threats to DFD elements . . . . .	21
4.3 Threat elicitation and documentation . . . . .	21
4.3.1 General assumptions . . . . .	21
4.4 Prioritizing threats . . . . .	24
4.4.1 Examples of privacy-risk assessment using DREAD . . . . .	24
4.4.2 Next phase after threat evaluation . . . . .	26

<b>5</b>	<b>Source Code Examination and Vulnerability Assessment</b>	<b>27</b>
5.1	User authentication . . . . .	27
5.1.1	SSO . . . . .	28
5.1.2	LDAP . . . . .	28
5.1.3	Internal login . . . . .	28
5.2	Publicly available DEV and STG environments . . . . .	29
5.2.1	Production data in publicly accessible environments . . . . .	29
5.2.2	Use of HTTP for communication . . . . .	30
5.3	JWT token forgery . . . . .	31
5.3.1	JSON Web Token . . . . .	31
5.3.2	CTU Survey JWT misconfiguration . . . . .	32
5.3.3	PoC exploit . . . . .	33
5.4	Separating student’s identity from Survey responses . . . . .	33
5.5	User actions revealed within Sentry breadcrumbs . . . . .	36
5.6	Identifiability, Unawareness and Non-compliance . . . . .	37
5.7	Insufficient data minimization in the database . . . . .	37
<b>6</b>	<b>Evaluation of Findings and Mitigation Strategies</b>	<b>41</b>
6.1	LINDDUN solution space . . . . .	41
6.1.1	Concealing association . . . . .	42
6.1.2	Guarding association . . . . .	42
6.2	Privacy enhancing strategies for the CTU Survey . . . . .	42
6.2.1	Enhancing respondent awareness . . . . .	43
6.2.2	Restricting access to deployment environments . . . . .	44
6.2.3	Ensuring GDPR and other privacy-related compliance . . . . .	45
6.2.4	Minimizing data stored in the database . . . . .	45
6.3	Suggestions for enhanced anonymity . . . . .	45
6.4	Recommendations for students . . . . .	47
6.4.1	How does the application ensure anonymity? . . . . .	47
6.4.2	Privacy recommendations . . . . .	48
<b>7</b>	<b>Conclusion</b>	<b>49</b>
<b>A</b>	<b>Database ER model [6]</b>	<b>51</b>
<b>B</b>	<b>Risk evaluation tables</b>	<b>53</b>
<b>C</b>	<b>Contents of the electronic attachments</b>	<b>57</b>
	<b>Bibliography</b>	<b>59</b>

## List of Figures

1.1	The infrastructure of the CTU Teaching Survey application [9]. . . . .	7
1.2	Illustration of the databases synchronization process. . . . .	8
4.1	Context model DFD of the CTU Survey . . . . .	20
4.2	Level 1 DFD of the CTU Survey . . . . .	20
4.3	Linkability of a data store threat tree from [20]. . . . .	23
5.1	Password setting page . . . . .	30
5.2	JWT header . . . . .	31
5.3	JWT payload . . . . .	32
5.4	JWT signature [32] . . . . .	32
5.5	Example of a JWT [32] . . . . .	32
5.6	JWT forgery: Generating teacher tokens in the STG environment . . . . .	34
5.7	JWT forgery: Login into the PROD version as another user (student) . . . . .	34
5.8	JWT forgery: Token forgery in the request's Authorization header . . . . .	35
5.9	JWT forgery: The course instructor page before and after token forgery . . . . .	35
5.10	Sentry error report . . . . .	38
5.11	Default settings for the text-based survey questions. . . . .	39
6.1	Taxonomy of LINDDUN mitigation strategies [1] . . . . .	41
6.2	Awareness notification: "Only 1 student from year 4 found in this course" [7] . . . . .	43
6.3	Awareness notification: JS code snippet . . . . .	44
6.4	DFD of an anonymous survey system from [4] . . . . .	46

## List of Tables

3.1	DFD symbols used for threat modeling [14]. Symbols from [17] . . . . .	14
4.1	Mapping privacy threat categories to DFD elements [14] . . . . .	21
4.2	Mapping privacy threats to the CTU Survey DFD elements . . . . .	22
4.3	DREAD risk rating table [22] . . . . .	24
4.4	Excerpt from the DREAD risk rating table B.1 . . . . .	25
4.5	Excerpt from the privacy-risk evaluation table B.2 . . . . .	25
B.1	DREAD risk rating for each threat applicable to Anketa CTU . . . . .	53
B.2	Ordered list of threats based on risk . . . . .	54

*First and foremost I would like to thank my supervisor Ing. Josef Kokeš, Ph.D. for his guidance and support throughout my work on the thesis. I would also like to express my gratitude to Ing. Michal Valenta, Ph.D. for giving me his permission and access to the Anketa CTU source code and database as well as for his transparency and consultations. Finally, I would like to thank my family and my partner who supported me throughout my studies.*



## Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 11, 2023

.....

## Abstract

The bachelor's thesis deals with an anonymity analysis of the Anketa CTU application, an online survey system used by the Czech Technical University for an anonymous course and instructor evaluation. The application falls into the category of anonymous survey systems, meaning the data contained in the survey questionnaires submitted by students should not be linked to the student's identity. The thesis examines this property, its implementation within the application, and discusses possible threats and pitfalls that could potentially compromise the users' privacy. The chosen approach for the anonymity analysis involves a threat analysis using the LINDDUN privacy threat modeling framework. The threat modeling process consists of studying and describing the application using both the publicly available information as well as the source code files and other parts of the system which are not accessible to the general public. After the information gathering phase, the next step is to create data flow diagrams of the system which depict individual system components and data flows. The individual parts of the system, as well as the system as a whole, are then examined for privacy threats that fit into any of the LINDDUN threat categories. The listed threats are then evaluated by their impact on the students' anonymity. The thesis describes the application's inner workings as well as a JWT misconfiguration and other vulnerabilities discovered within the system. Finally, the thesis proposes suitable mitigation strategies and anonymity-enhancing solutions.

**Keywords** online survey system, Anketa CTU, anonymity analysis, privacy threat modeling, LINDDUN, JWT

## Abstrakt

Tato bakalářská práce se zabývá analýzou anonymity aplikace Anketa ČVUT, webové aplikace, kterou využívá České vysoké učení technické pro interní hodnocení výuky. Tato aplikace spadá do kategorie anonymních anketních systémů, což znamená, že data obsažená v dotaznících odevzdaných studenty by neměla být spojitelná s jejich identitou. Tato práce zkoumá především tuto vlastnost, její implementaci v rámci aplikace, a předestírá možné hrozby a nedostatky, které mohou ohrozit soukromí uživatelů. Analýza anonymity je v práci pojata jako analýza hrozeb s využitím modelu LINDDUN jakožto hlavní metodologie pro modelování hrozeb soukromí. Proces analýzy hrozeb spočívá ve studiu a popisu analyzované aplikace, a to jak pomocí veřejně dostupných informací, tak i pomocí zdrojových kódů a dalších částí systému, ke kterým nemají přístup běžní uživatelé. Po sběru informací následuje vytvoření diagramů datových toků systému, které zobrazují jednotlivé komponenty systému a datové toky mezi nimi. Jednotlivé části systému a celý systém jsou poté prozkoumány z hlediska možných hrozeb soukromí, které spadají do některé kategorie hrozeb obsažené v metodologii LINDDUN. Zjištěné hrozby jsou následně hodnoceny z hlediska dopadu na anonymitu studentů. Tato práce popisuje vnitřní fungování aplikace, stejně jako konkrétní bezpečnostní hrozby, jako například chybnou konfiguraci JWT a další zranitelnosti nalezené v rámci systému. V závěru práce jsou navržena vhodná opatření ke zlepšení anonymity a ochrany soukromí uživatelů.

**Klíčová slova** online anketní systém, Anketa ČVUT, analýza anonymity, modelování hrozeb soukromí, LINDDUN, JWT

## Abbreviations

ANKOM	Anketa CTU commenter role
ANNAH	Anketa CTU viewer role
ANNVED	Anketa CTU management role
ANSF	Anketa CTU faculty administrator role
ANSS	Anketa CTU system administrator role
API	Application Programming Interface
CIA	Confidentiality, Integrity, Availability
CIC	Computing and Information Centre
CTU	Czech Technical University
DFD	Data Flow Diagram
DREAD	risk assessment model
ECDSA	Elliptic Curve Digital Signature Algorithm
FIT	Faculty of Information Technology
GA	Google Analytics
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IoI	Item of Interest
IS	Information System
JS	JavaScript
JSON	JavaScript Object Notation
JWT	JSON Web Token
KOS	the main CTU IS
LINDDUN	privacy threat modeling framework
MITM	man-in-the-middle attack
MUC	misuse case
OS	Operating System
PET	Privacy Enhancing Technique
PHP	scripting language
PII	Personal Identifiable Information
PK	Public Key
PoC	Prove of Concept
RSA	Rivest–Shamir–Adleman public-key cryptosystem
REST	Representational State Transfer
SDL	Secure Development Lifecycle
SSH	Secure Shell
SSO	Single Sign On
STRIDE	security threat modeling framework
UI	User Interface

# Introduction

The subjects of an individual's privacy and anonymity have never been more significant than in today's digital age.

Privacy has become a key concern in various online systems. The topic of protecting the users' privacy is crucial in highly sensitive areas such as healthcare as well as in systems that we consider a part of our everyday life. For example, social media platforms in which issues such as user's unawareness of the consequences of sharing personal data with the system or an insecure or non-compliant way in which the system deals with the user's personal data could result in breaching the user's privacy.

A distinct group of systems, where user's privacy and anonymity are a mandatory requirement, are anonymous voting and feedback systems. This group includes online voting systems, anonymous polls and survey applications as well as whistleblowing systems. The key concern for these systems is to ensure the separation between the user's identity and their submitted ballots (or other actions that the system ensures to keep anonymous). It is the responsibility of the system to ensure that its users cannot be held accountable for anonymous actions they take within the system. [1]

This thesis deals with an anonymity analysis of the CTU Teaching Survey (or Anketa CTU) which is an online teaching survey system used by the Czech Technical University for an internal evaluation of teaching by collecting feedback from students. It also examines and describes the application's inner workings, focusing particularly on the way in which the system ensures the students' anonymity.

The main objective and contribution of this work is in heightening the transparency of the system's processes which could contribute to improvements in the system architecture and implementation, enhanced user awareness as well as respondents' trust in the system and higher survey response rates.

The first objective was to study the application by gathering publicly accessible information about the system's technical details as well as by studying the application source code and database. The second thesis objective (which was also the next step of the anonymity analysis) was to use the previously collected information to create a model of the system and conduct a threat analysis of the system with a particular focus on threats which could result in compromising the survey respondent's anonymity.

Threat modeling is a structured approach to eliciting potential pitfalls in software systems. It is a well established technique for discovering security threats. A significant amount of literature has been written that deals with the subject of threat modeling for security. One notable example is the work of Adam Shostack, one of the developer's of Microsoft's STRIDE, who's "*Threat modeling: Designing for Security*" is considered as the most influential text on the topic of security threat modeling. [2, 1]

The abundance of literature discussing threat analysis for security could be attributed to the

fact that security engineering is (relative to privacy engineering) an older and well established field. While security is indispensable for privacy, since it ensures confidentiality, integrity and availability of the whole system and all data processed by the system, the requirements for preserving user's privacy and anonymity might differ from security requirements. [1] For example, there is a difference between confidentiality and anonymity. While an anonymous survey system ensures that the collected data is not linked back to the original respondent's personal identifiers, a confidential survey system might store the survey responses together with the respondent's identifiers and only ensure that the information is stored securely, preventing unauthorised personnel from accessing the information.

Since anonymity is a privacy property and the CTU Survey is by design an anonymous online survey system, the conclusion was to conduct a privacy threat analysis (as oppose to a security threat analysis) of the system. The privacy threat modeling framework LINDDUN was selected as the main method used for the analysis.

The first chapters of the thesis are dedicated to first introducing the CTU Teaching Survey application, its purpose, history, infrastructure and technologies used as well as other defining properties such as roles that exist within the system and the timeline of a single survey instance. The following chapter introduces the theoretical concepts and definitions related to anonymity and privacy and also discusses their importance in the context of an anonymous online survey system such as the CTU Survey. The focus in the third chapter is on introducing the concept of threat modeling and available threat modeling methods. This chapter also introduces the privacy threat modeling methodology LINDDUN [1] and includes definitions of the privacy threat categories encapsulated in the LINDDUN acronym (Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of Information, Non-compliance).

The second part of the thesis deals with the practical analysis of the CTU Survey. The information previously gathered about the system as well as my understanding of the application source code and database were leveraged to first create a threat model of the application. The following sections describe the process of applying the LINDDUN threat modeling methodology to the CTU Survey, as well as the process of threat evaluation using the Microsoft's DREAD model. In Chapter 5 the focus shifts from the model of the application to the actual system. The application is examined for possible manifestations of threats in forms of vulnerabilities using the ordered list of threats created during the threat modeling phase. The chapter also presents the discovered vulnerabilities.

The aim of the last chapter is to propose suitable mitigation strategies for the privacy threats applicable to the CTU Survey system. The recommendations are derived mostly from the LINDDUN solution space [3], as well as from the study "*An information system architecture for ensuring anonymity of student survey responses*" which was published in 2019 by a group of researches from the Old Dominion University and which describes a possible way of designing an anonymous online survey system which guarantees a complete anonymity of student responses. [4] The recommendations are targeted both at the system stakeholders (designers, developers, testers and administrators), as well as the application users and survey respondents.

# The CTU Survey Application Overview

*The aim of the first chapter is to provide a high-level overview of the CTU Survey application, including its purpose, timeline of a single survey instance, history and technologies used. The last section of this chapter includes an overview of the underlying infrastructure as well as of all deployment environments used by the CTU Survey development team.*

## 1.1 Purpose of the CTU Survey

All public universities in the Czech Republic are required to conduct regular internal evaluation of their teaching. [5] How exactly is this requirement fulfilled is determined by each university independently. The CTU Teaching Survey (or Anketa CTU) is a web application used by the Czech Technical University for that purpose. It is an online solution for collecting and evaluating feedback on teaching from students of the CTU. It has been developed and maintained by the CTU Computing and Information Centre (CIC), with the support of the CTU Internal Projects and it is described in detail in the methodological guideline no. 3/2022. [6]

## 1.2 Survey modules

The CTU Survey consists of three modules – the *filling and displaying* module, the *administration* module and the *reporting* module.

The filling and displaying module, as its name suggests, is a web application which implements the main functionalities of the CTU Survey. Students and teachers can log into the application through its graphical user interface (GUI) and interact with it. Within the module they can fill in and submit survey questionnaires as well as view and comment on survey results.

The administration module is the tool for administering individual faculty surveys. This application is not accessible to all students and teachers. It is meant primarily for faculty survey administrators.

The reporting module is the tool for displaying reports and general information about surveys (information such as what kinds of questions appear in the survey questionnaires, or how many students participated in each survey). It is publicly accessible from the main page of Anketa CTU in a form of statically generated HTML pages.

This thesis focuses on the filling and displaying module. The administration module and the reporting module are out of scope of the anonymity analysis.

### 1.3 System roles

*System administrator (ANSS)* is responsible for the entire lifecycle of a survey. Their responsibilities include preparing data for upcoming surveys, setting roles of faculty administrators and assisting the faculty administrators with tasks which they are not authorised to handle in the system. [6]

Each faculty survey has its *faculty administrator (ANSF)*. They are authorised to modify data before the launch of a survey. That includes courses and course instructors with initial data transferred from the IS KOS system. The ANSF also have the right to make decisions and manage certain events within the Survey lifecycle. [6]

*Student* is the survey respondent. [6]

*Teacher* is a role assigned to an individual, who participated as a course instructor in at least one of the CTU courses opened in the given semester. [6]

User with the *management (ANVED)* role is given the right to respond to students' comments in the survey comment section for any course, and to view survey results before they are available to students. [6]

*Commenter in course non-specific survey (ANKOM)* is the role which allows its holder to comment on student feedback during the evaluation phase of the closed non-subject survey. [6]

Finally the *viewer (ANNAH)* role allows its holder to view the survey results. This role is attributed to students, teachers and all the other roles listed above. [6]

### 1.4 Lifecycle of a survey instance

The CTU Survey is launched twice during each academic year, at the end of each semester. A single survey instance consists of four phases: *preparation, filling of the survey, evaluation and publication*. [6]

The first phase is the preparation phase. In this initial phase the survey system administrator prepares data for the upcoming surveys (which are retrieved from the CTU KOS system) and sets the roles of faculty administrators. It is then up to the faculty administrators of each faculty survey to conduct a final check and possibly make adjustments before launching, and finally launch the surveys. [6]

The second phase usually takes place during the exam season. At this stage surveys are open for students who can provide feedback on the courses they have already completed. As soon as a student completes a university course the corresponding questionnaire appears in their CTU Survey account. Students are then able to view the questionnaire through the application's GUI and submit ratings and opinions about the course and its instructors. Ultimately, at the end of the exam season, students are enabled to rate all of their assigned courses, regardless of whether they were successful in completing them or not. [6]

The CTU Survey is an anonymous survey system. This means that the data contained in the survey questionnaires submitted by students are not linked to the students' identity (as opposed to confidential survey systems, in which data is linked back to personal information). [7]

Answers submitted by students are meant to be anonymous by default. The students' identity is separated from their survey questionnaire during submission. However, students have the option to disclose additional information about themselves – namely their name, study major, year of study, average score and role of the course within their study branch. Students are also free to disclose more information in answers to the open-ended text-based questions.

The CTU Survey ensures that students are able to fill in and submit only one questionnaire for each of the courses they were enrolled in and that students are not able to rate courses they did not sign up for. This is meant to ensure the relevancy of the survey results. [7]

All answers are saved in the survey database. In the third phase, the survey closes for students and survey results are examined and evaluated by the CTU teachers and management. In this



phase, teachers can view and comment on the answers submitted by students. They can also decide to moderate inappropriate comments. [6]

The final phase is the publication of results. At this point the results are accessible to all students and internal and external teachers of the CTU [6] (anyone, who possesses a valid CTU identity or set of credentials).

It is important to point out that the content presented in the Anketa GUI is tailored to each individual user. Both students and teachers are able to view survey results only for courses relevant to them and are given different options and rights, which they can execute within the application, depending on their role (teacher or student role). For instance, a student from the Faculty of Architecture can view survey results for courses offered within their faculty but not for courses in other faculties. The same student would be able to see comments made by a course instructor. The instructor might see a button allowing them to alter or delete their own comments, whereas the student can only view the teacher's comments and is not given the right to modify them.

## 1.5 History

The first CTU online survey system was created in 2002 at the Faculty of Electrical Engineering. It was meant to provide a unified framework for the faculty self-assessment. The first version was written in PHP. The application turned out to be successful and was adopted by other faculties in 2004. However, it soon became apparent that the current concept of the application was not able to cover all of the newly emerging requirements. Robert Jiřík therefore implemented a new version of the application in the year 2007, this time written in Java. The design of his new version was easily extensible and gave an option to add a GUI interface in English for foreign students. His version was further developed and maintained by the development team lead by Ing. Michal Valenta, Ph. D., consisting of CTU students Lukáš Frélich, Vladimír Kobětič, Josef Sin and Stanislav Šimek. [7]

The application was continuously enhanced with new features. For example in 2014 a new functionality was added to the application which allowed students to add their names to their survey responses. Before this change, the survey questionnaires were fully anonymous (separating students' personal identifiers from their responses) and it was only possible to sign the questionnaires by typing the student's name within the text based fields. However, that way the authenticity of the author of the survey response was not guaranteed. The new feature, when activated, saved the student's name together with their student identification number to the survey database and later displayed it together with the student's survey answer on the application GUI. This feature was meant to ensure the authenticity of the respondent. [7]

Due to the increasing dissatisfaction of students with the application, an extensive survey was conducted at the Faculty of Information Technology (FIT) in 2016. Its outcome was a proposal for a comprehensive change in the user interface. However, the proposed change would be difficult to implement within the current data model of the application. Therefore, a new design was proposed in 2018 by another CTU student, David Knap, who, in his master's thesis, took on the task of defining the requirements and designing the new interface for the CTU Survey 3.0. [7]

His prototype was accepted by the CTU students and teachers and became the baseline for Anketa CTU 3.0. In the following years several CTU students (mostly from the Faculty of Information Technology) contributed to the Survey's further development. In 2019, Vojtěch Štecha followed up on the work proposed in David Knap's diploma thesis and implemented the filling and displaying parts of the application. [8]

In 2020, a new application for the administration of individual faculty surveys was developed by Jakub Jun in collaboration with Duc Thang Nguyen. [9]

There is currently a new version of the frontend of the CTU Survey in development. It was designed by Nam Nguyen Hai in 2021 in his bachelor thesis and it was meant to possibly replace

the current application frontend in the future. [10] However, in my thesis I analyse the filling and displaying module of the CTU Survey version 3.0.5, which is still, as of summer term 2023, deployed in production and used by the CTU students and teachers.

## 1.6 Technologies used

The filling and displaying module of the CTU Survey version 3.0 is a web-based application with a client-server architecture.

The frontend is written in JavaScript (JS) using the React library and the Bootstrap framework. It implements the client side of the application and provides a GUI for users to interact with the application. The React library provides components which receive data and return what should appear on the screen. It adds interactivity, but React by itself was not built to handle all functional aspects such as routing, state management and communication with the API server. These functionalities are implemented using other components from the React ecosystem. The React-Redux library is used for state management, React-Router for routing (navigating on the website) and the React-Thunk is used as a middleware to allow communication with the API server alongside with the Axios library. [8]

The server side was written in Java using the Spring framework. It implements the application logic and provides a REST API. The CTU Survey client uses the REST API to fetch data and to retrieve information about the current application state from the server. The server also takes care of the persistency and connects to the Survey database through a JDBC driver. [8]

The database is Oracle. The database model was designed mainly by Ing. Michal Valenta, Ph.D., and it is publicly available to view either in the official Anketa CTU system documentation or in the diploma thesis of Vojtěch Štecha. [6, 8] For the convenience of the reader, the database ER model is also added as the Attachment B of the thesis.

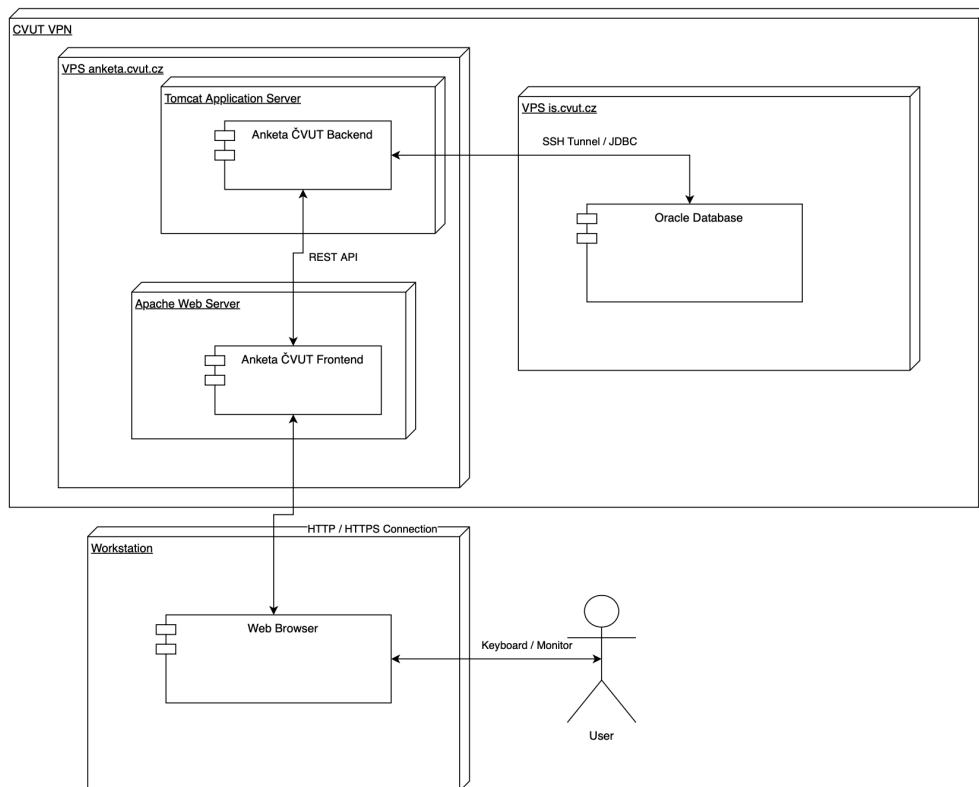
## 1.7 Infrastructure

The following section describes the underlying infrastructure of Anketa CTU. I obtained the technological description and the model of the infrastructure from Jakub Jun's bachelor thesis. [9] The original image described the state of the application in 2020. However, the application's administrator, Ing. Michal Valenta, Ph.D. confirmed the accuracy of the original model as well as all information about used technologies.

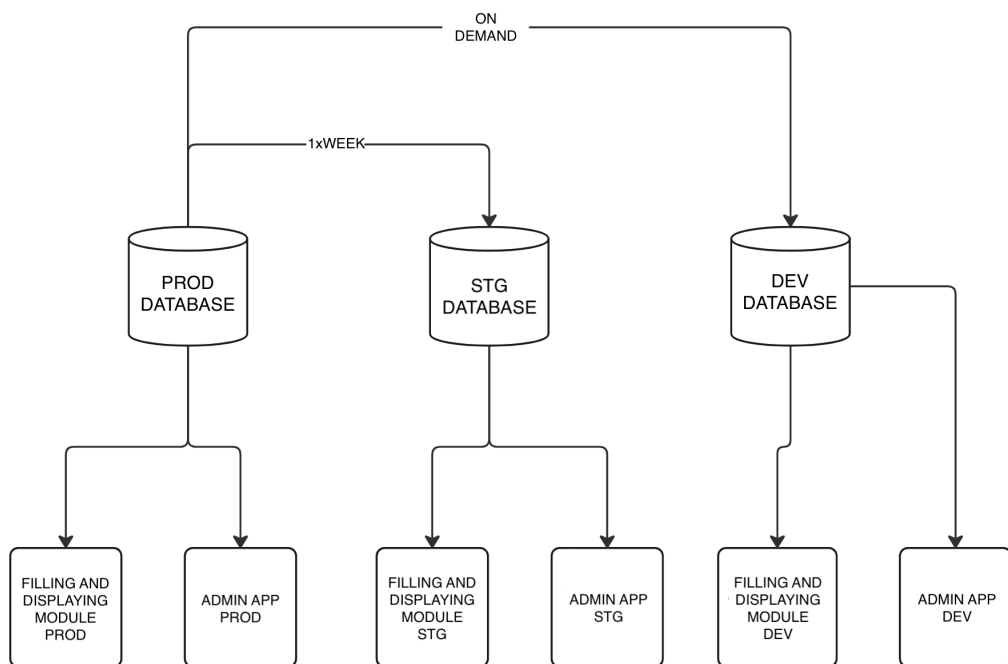
Both the client side and the server side of the filling and displaying module run on the same virtual machine with a Linux OS. There is an Apache Web Server for the frontend and Tomcat Application Server running the application backend services. The Oracle database is located on a separate virtual machine. For security reasons, all of the sensitive parts of the system, such as the database and servers, are located within the CTU virtual private network (VPN). Communication between the system components is secured through an SSH tunnel. [9]

Two new deployment environments, development (DEV) and staging (STG), were added to ease the process of development. Latest versions of the application are deployed to the DEV environment. The STG environment is meant for a final check and testing of bigger changes before their deployment to production. [9]

There are two additional databases, one for each of the corresponding environments. These databases are identical in structure with the production (PROD) database. The STG database is synchronized with the PROD database once a week, whereas the data from the DEV version are copied irregularly on demand. The process of database synchronization is illustrated in figure 1.2.



■ **Figure 1.1** The infrastructure of the CTU Teaching Survey application [9].



■ **Figure 1.2** Illustration of the databases synchronization process.

# Anonymity

*The aim of the following chapter is to introduce, define and explain one of the core theoretical concepts of this thesis, anonymity, and its importance in the CTU Teaching Survey, as well as other privacy related terms such as plausible deniability, unlinkability, undetectability and unobservability.*

## 2.1 Definition of anonymity

Anonymity is generally well understood by most people, meaning that people have an intuitive understanding of the concept of anonymity and what it means to be anonymous. However, to avoid any confusion in later chapters and to establish a common understanding with the reader, it is important to provide a clear definition of the term. In this thesis, the term anonymity and other related terms regarding an individual's privacy are defined according to the work of Andreas Pfitzmann and Marit Hansen, who aimed to provide a unified, expressive, and clear terminology for discussing privacy in cyberspace. [11]

Their terminology was developed in the setting of *entities (subjects and objects)* and *actions*. A subject could be a human being, a legal person, or a computer. Objects are messages which can be sent or received by subjects (*senders* or *receivers*) through a communication network. An *attacker* or an *adversary* is an entity or a set of entities working against a protection goal such as anonymity. The adversary may be an outsider excluded from the communication or an insider able to participate in normal communications. [11]

***Anonymity of a subject from an adversary's perspective means that the adversary is not able to sufficiently identify the subject within a set of subjects, the *anonymity set*.*** [11]

This definition implies that anonymity is not a binary property, it is not a property which the subject either does or does not possess, but that it is rather a quantifiable property and that there might be a need to define a threshold for where a subject's anonymity begins in a specific setting and with regards to a specific attacker. [11]

There are therefore multiple variables which can impact the anonymity of a subject — the size of an anonymity set being one of them. The larger the anonymity set, the more difficult it might be for an attacker to sufficiently identify a specific subject. Other variables are the motivations and skills of the attacker, or the attacker's role and their set of privileges within the system.

## 2.2 Privacy vocabulary

Anonymity as a privacy protection goal closely relates to other privacy properties such as *plausible deniability*, *pseudonymity*, *unlinkability*, *undetectability* and *unobservability*. [11]

A *pseudonym* is an identifier of a subject other than the subject's real name. Pseudonymity then refers to the use of pseudonyms as identifiers. Pfitzmann and Hansen classify pseudonyms into three categories based on the link between the pseudonym and its holder: *public pseudonyms*, which may be linked to the holder from the beginning, *initially non-public pseudonyms*, in which the link may be known by certain parties, but is not initially public and *initially unlinked pseudonyms*, in which the link between the pseudonym and its holder is not known to anybody. [11] For example, in the context of the CTU Survey the CTU usernames serve as pseudonyms. These pseudonyms could be classified as public pseudonyms since they are all derived from their holders' names and surnames and are therefore strongly linked to the holder's identity and are either publicly known or easily deducible.

Unlinkability of two or more items of interest (IoI) from an adversary's perspective means that within the system, the adversary is not able to sufficiently distinguish whether these IoIs are related or not. IoI could be an entity (e.g. sender of a message or the message itself), an action (e.g. the act of sending or receiving a message) or an identifier (e.g. a name). [11]

Undetectability of an IoI from the adversary's perspective means that it is impossible for the adversary to sufficiently distinguish whether the IoI exists or not. [11]

And finally unobservability of an IoI means, that the IoI is undetectable by all subjects uninvolved in it and the subjects involved in the IoI are anonymous to all subjects both involved and uninvolved in the IoI. [11]

Unobservability is stronger than anonymity and undetectability, since it reveals only a subset of the information anonymity and undetectability, with regards to the same attacker. We can say that unobservability implies anonymity and also that unobservability implies undetectability. [11]

## 2.3 Privacy and security

At this point, it is important to note the distinction between the terms *data security* and *privacy*. In the digital world, data security refers to safeguarding of data from unauthorized access, often-times involving protection measures against potential breaches or leaks. It is the act of keeping data secure, and ensuring it is not accessed by unauthorized subjects. Privacy on the other hand is a concept related to users' personal identifiable information (PII). It concerns the protection of data, which can be linked to a natural person. Privacy is the right of a subject to have control over how their personal information is collected, stored and used. [12, 1]

It is difficult to ensure subjects' privacy without considering security, since it is necessary to have a secure system which ensures the CIA triad, confidentiality, integrity and availability of all assets, to be also able to protect the user's data and PII. However, security requirements and privacy requirements might differ in certain situations and systems and sometimes they might even contradict each other. Plausible deniability is an example of a privacy property which can be seen as a security issue in one context and as a privacy requirement in another. It refers to the subject's ability to deny having performed an action. From the adversary's perspective, plausible deniability means that the adversary is not able to prove that the subject knows, has done or has said something. It ensures that the subject cannot be held accountable for their actions and it is therefore a desirable property in systems such as online anonymous voting systems or whistleblowing systems. There are however many systems which demand non-repudiation, the opposite of plausible deniability, to ensure user accountability (e.g. for e-commerce applications). [1]

## 2.4 Importance of anonymity in Anketa CTU

The CTU Survey 3.0 was designed as a non-anonymous application which implements an anonymous survey system as one of its core features. This means that to log into the filling and displaying module, users have to provide a valid set of credentials consisting of the CTU username and password.

Once a user logs in, a session is initiated, granting access to various functionalities of the website. These include viewing reports of previously conducted surveys, submitting feedback on specific courses and instructors, providing general faculty feedback, and commenting on survey responses submitted by students. The specific actions which the user can perform are dependent on the user's role and the current survey phase. [6]

Furthermore, the application includes a Slack feedback gathering tool that enables users to report bugs, express opinions, and suggest improvements to the system. It's important to note that among all of these features, only the submission of survey questionnaires was designed to provide a certain level of anonymity.

For his master's thesis, David Knap conducted comprehensive surveys and discussions among CTU students, individual faculty members and faculty survey administrators to gather their opinions on the new version of Anketa 3.0. The anonymity of the survey system turned out to be a controversial topic, mostly because of opposing opinions about the extent to which the student survey should be anonymous. [7]

According to Knap's survey results, a majority of the participating students (80 %) indicated a preference for optional anonymity, whereby the default setting of the system would be fully anonymous but the respondents could choose to add their names. 18 % of students expressed a preference for a fully anonymous survey, while only 2 % of students were in favor of a fully transparent survey system that would include and publish the student's PII with each survey response. [7]

The opinions of individual CTU teachers and management regarding the newly designed survey system differed from the general student opinion. Some teachers stated that they would prefer a confidential or semi-anonymous teaching survey rather than an anonymous survey. Their argument was that they would like to know at least some information about the survey respondent, such as their average grade, to determine the relevancy of the content of their qualitative survey responses. Others expressed their concern about the possibility of respondents re-identification associated with the use of a semi-anonymous survey. In situations where the combination of mandatory information included with a student's response is unique to them among all students enrolled in the course, always including and publishing this information could potentially result in the student being identified. [7]

David Knap's work also addressed the relatively low response rate of the survey and the reasons that deter CTU students from providing feedback. According to the Vice-Dean for Education, one of the reasons for students' reluctance to fill out the survey was their lack of trust in the system's anonymity and negative experiences with subsequent retaliation by course instructors. The proposed solution was to clearly document the principles of anonymity in the survey system, present the information in a way that is understandable even to those without technical knowledge, and to make the description publicly available along with information about the survey's existence, for example, in materials created for the first-year students at CTU. [7]

The final solution which was chosen regarding the level of anonymity within the CTU Survey 3.0 was the solution that was most preferred by students, that is to implement a survey system which is anonymous by default but which allows adding certain student attributes (such as their name, year of study or average grade) alongside their qualitative answers in the form of a so called *anonymity settings panel*. [7]





# Threat Analysis

*The following chapter introduces the process of threat modeling for security and privacy. It also introduces and describes the LINDDUN privacy threat modeling framework, which was selected as the main method for the anonymity analysis of the CTU Teaching Survey.*

## 3.1 Threat modeling

Threat modeling generally consists of analyzing system representations to bring out concerns about security and privacy characteristics. It involves assessing the current state of the system, asking questions that help to identify design and implementation issues, potential pitfalls and threats, evaluating those threats and determining the most suitable ways to mitigate them. [13]

In the context of software development, threat modeling means creating a model of the system at hand and analyzing it for potential risks, to essentially see what can go wrong in the system and what could be done to prevent it.

Threat modeling of a system is a structured approach which can be split into three high level steps: decomposing the application, determining and ranking threats, and determining countermeasures and mitigation. [14]

It is best to incorporate threat modeling from the early stages of the development of a software project and then apply it continuously during the development lifecycle (SDL). However, threat modeling conducted in combination with source code examination on existing applications outside the SDL helps to clarify the complexity of source code analysis and promotes a depth-first approach as opposed to a breath-first approach. This means that it enables the analyst to prioritize the security code review of components where threat modeling indicates a higher risk. [15]

### 3.1.1 Decomposing the application




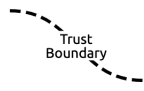

The aim of this first step of the analysis is to gain an understanding of the application and how it interacts with external entities. This is achieved through information gathering, documentation and modeling of the system. [15]

Data Flow Diagram (DFDs) of the application are created to provide a visual representation of how the application processes data at varying levels of abstraction. A DFD contains all essential components of the system as well as external entities which interact with the application, maps different trust boundaries and entry points and showcases how data flows through the system. [14] There are various tools available, both open-source and proprietary, that can be used to create a DFD of an application. Examples of popular open-source threat modeling tools include

Cairis, Microsoft Threat Modeling Tool or the OWASP Threat Dragon. [16]

I have depicted the various symbols used in DFDs for threat modeling, along with their name and description, in table 3.1. I obtained the images of the symbols used in table 3.1 from the OWASP Threat Dragon modeling tool. [17]

■ **Table 3.1** DFD symbols used for threat modeling [14]. Symbols from [17]

Symbol	Name	Description
	Actor (or external entity)	The actor symbol represents any entity outside the application that interacts with the application through an entry point.
	Process	The process symbol represents a unit that handles data within the application. The unit may process the data or perform an action based on the data.
	Store	The store symbol represents a data store. Data stores only store data, they do not modify them.
	Trust boundary	The symbol for trust boundary (or privilege boundary) is used to separate different trust levels as the data flows through the application. Boundaries show any location where the trust level changes.
	Data flow	The data flow symbol is used to illustrate the data movement within the application. An arrow at the end of the line suggests the direction of the data flow.

### 3.1.2 Determining and ranking threats

The second step in the threat modeling process involves identifying and prioritizing potential threats. To ensure that threats can be identified in a structured and repeatable manner, it is recommended to adopt an established threat categorization framework that includes a set of defined threat categories along with relevant examples. One widely recognized threat modeling framework is STRIDE, which was created by Microsoft and is considered a well-established security threat modeling methodology. [1] The acronym STRIDE represents the attacker goals

of Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privileges. [18]

There are other available threat modeling methodologies such as PASTA or LINDDUN. [18] After evaluation of available methods and consulting with my supervisor, I decided to utilize the LINDDUN methodology for assessing the threats to Anketa CTU anonymity.

LINDDUN is a privacy-focused threat modeling methodology. It was inspired by STRIDE and it is similar in certain aspects — just like STRIDE, LINDDUN is a model-based approach that leverages data flow diagrams (DFDs) as representations of the system under analysis. The main difference between the two approaches is that LINDDUN was designed specifically as a privacy threat modeling methodology. Its focus is solely on threats which are relevant to subjects' privacy. [1]

LINDDUN, like STRIDE, is designed to be integrated into the SDL and is compatible with the *Privacy by design* paradigm which states that privacy should be embedded in the early stages of the SDL. [1]

Threats from the threat categories of the chosen threat modeling framework are then applied to all system elements listed in the DFDs. The security risk for each threat can be determined using a value-based model such as the Microsoft's DREAD. [15] The process of ranking threats by risk helps to prepare the scene for the final step of the threat modeling process, which is the assessment and proposition of effective mitigation strategies. [14]

### 3.1.3 Countermeasures and mitigations

The purpose of the final step of the threat modeling process is to determine the protective measures that can prevent a threat from being realized. [15]

Vulnerabilities may be mitigated by implementing countermeasures. Such countermeasures can be identified with the help of threat-countermeasure mapping lists. The threats are sorted by their risk ranking assigned in step two of the threat modeling process from the highest to the lowest. The ordered list of identified threats is then used to prioritize mitigation efforts. [14]

## 3.2 LINDDUN framework

The LINDDUN privacy threat modeling methodology was first published by the DistriNet Research Group at KU Leuven in 2010. It adheres to the *data protection by design and by default* approach imposed by the General Data Protection Regulation (GDPR) and provides a systemic methodology to assess privacy threats and their impact on systems in the context of software development. It was inspired by the security threat modeling methodology STRIDE and it was designed to be complementary to a security threat analysis. [19]

The framework provides a knowledge base of privacy threat types (which together form the LINDDUN acronym), mapping tables (to map threat types onto system components), threat trees, as well as a taxonomy of mitigation strategies (depicted in figure 6.1) and privacy enhancing techniques (PETs). [19]

### 3.2.1 Privacy threat categories

The acronym LINDDUN encapsulates the different privacy threat categories. It stands for Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness and Non-compliance. These privacy threat types are negations of the following privacy properties: Unlinkability, Anonymity, Plausible deniability, Undetectability, Confidentiality, Content Awareness and Compliance. [19]

**Linkability** is the opposite of unlinkability. It refers to the attacker's ability to sufficiently distinguish whether two IoIs are linked or not. It refers to the inability to hide the link between two or more actions/identities/pieces of information. [3]

Linkability on its own is not necessarily a privacy issue. It can, however, result in severe privacy issues when the linkable data leads to subject's identification or inference. An anonymity set can become smaller with additional information. When so much information is linked that the anonymity set consists of just one subject, this subject is identifiable. [3]

In the context of the CTU Survey, the anonymity set could be, for example, a group of students who all signed up for the same course. Course capacity at the CTU is generally limited to no more than a few hundred places and students are allowed to submit their survey questionnaires only for the courses they signed up for. Therefore, all students who submit their opinions in the survey on a specific subject are part of one anonymity set and could become, with additional information, potentially identifiable as individuals.

**Identifiability** as the opposite of anonymity refers to the attacker's ability to sufficiently identify the subject within a set of subjects (i.e. the anonymity set), or the inability to hide the link between the subject's identity and the IoIs. [3]

While anonymity refers to hiding the link between an identity and an action or a piece of information, identifiability is usually a consequence of linking data to the same subject. [3]

**Non-repudiation** means having an evidence concerning the occurrence or non-occurrence of an event or action. While this is usually a wanted property in security (with its opposite Repudiation being one of the threat categories in the STRIDE acronym), it could be seen as a privacy issue depending on the type of a system. [1]

Non-repudiation is related to plausible deniability. Non-repudiation and plausible deniability are mutually exclusive. Systems either require strong non-repudiation to ensure accountability or they require plausible deniability. The first category of systems could be represented e.g. by an e-shop where the vendor can use a signed receipt as evidence that the user received their item. For other types of applications, such as whistle-blower systems, online voting or anonymous survey systems, users may desire plausible deniability for privacy protection such that there will be no record to demonstrate the communication event, the participants and the content. In these scenarios, non-repudiation is a privacy threat. [3]

**Detectability** is the opposite of undetectability. It means that the attacker is able to sufficiently distinguish whether an IoI exists or not.

Examples of detectability include: knowing whether an entry in a database corresponds to a real person, being able to distinguish whether someone or no one is in a given location, knowing whether a message was sent, etc. [3]

**Disclosure of information** refers to the act of exposing information to someone who is not authorized to see it. It is the only threat category which occurs both in the LINDDUN and the STRIDE acronyms. [3]

**Unawareness** or Content unawareness indicates that the user is unaware of the consequences of disclosing information to the system. [19]

**Non-compliance** is the failure to follow the data protection legislation, the advertised policies or the existing user consents. [3]

### 3.2.2 Methodology

The three main steps of the LINDDUN methodology correspond to the previously introduced steps of a general threat modeling process: modeling the system using DFDs, systematically iterating over the DFD elements and exposing privacy threats associated with the elements, and finally managing the uncovered threats by finding suitable mitigation strategies. [19]

The following chapters introduce and explain the detailed steps of the LINDDUN methodology, as they were applied during the process of the privacy threat analysis of the CTU Teaching Survey.



# Threat Analysis of the CTU Survey

*This chapter presents the process of privacy threat modeling applied to the CTU Teaching Survey application. The final model was based on the system description and information gathered and presented in previous chapters. The system was decomposed into two levels of abstraction. The individual system components were examined for privacy threats from the LINDDUN framework and evaluated using the DREAD risk assessment model.*

## 4.1 Threat model

The tool of choice for creating the threat model of the CTU Survey was the OWASP Threat Dragon. [17] This modeling tool allows creation of DFDs as well as elicitation of threats from the STRIDE, LINDDUN and CIA threat categories. All processes, actors, or data flows that are not included in the scope of the anonymity analysis but that are either an important part of the system or interact with the application are marked with a dashed line.

The context model DFD (figure 4.1) represents a high level overview of the application. At this level the only components are the application itself, the actors who interact with it and the data flows between the application and the actors. The *0.0 Survey Web Application* process at this level represents a collection of subprocesses which were be further decomposed in the Level 1 DFD. The *0.0 Survey Web Application* process is connected to the actors of the system. The *1.0 User* actor represents users of the filling and displaying application, the *2.0 Survey Administrator* actor represents users of the administration application, the *7.0 external authentication servers* element represents the external authentication providers and finally the *8.0 Goggle Analytics* actor is a representation of the 3rd party Google Analytics service.

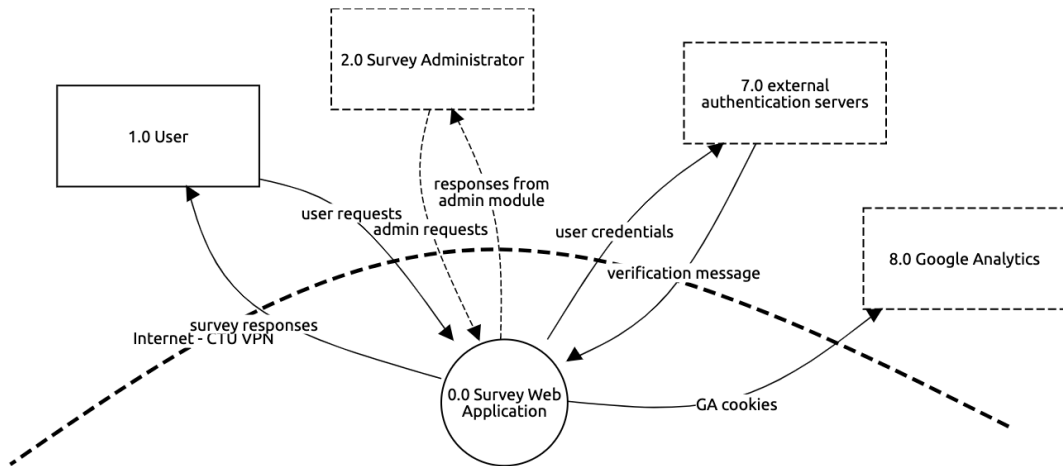
The level 1 DFD (figure 4.2) was based on the diagram of the system infrastructure from figure 1.1. The *0.0 Survey Web Application* process is further decomposed into the *5.0 Survey Administration App* process, the frontend (*3.0 Frontend Web Server*) and backend (*4.0 Backend App Server*) of the filling and displaying application and the database (*6.0 Oracle Database*),<sup>1</sup> and it includes internal trust boundaries within the CTU VPN. The *7.0 external authentication servers* actor is split into the *7.1 SSO IdP* and the *7.2 LDAP server* actors.

Privacy threat analysis deals primarily with the information which is collected by the system, how is the information stored and what information is shared with external entities. The analysis

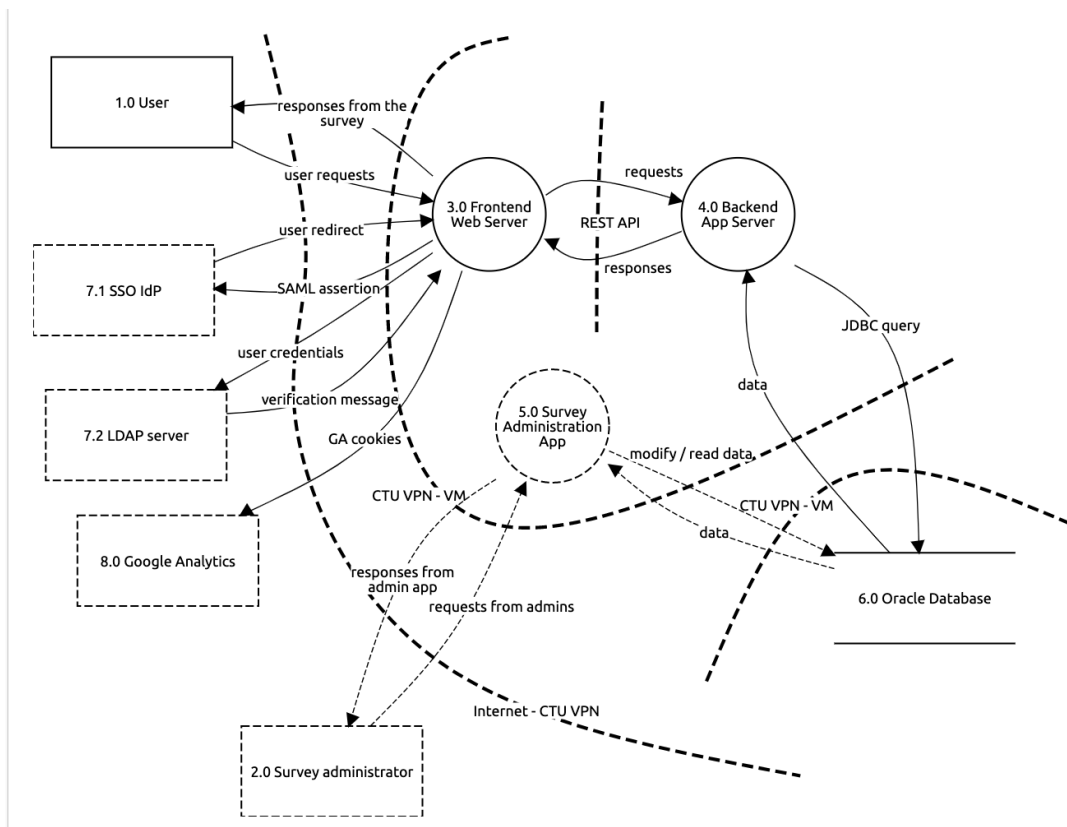
---

<sup>1</sup>The reporting module of the CTU Survey is a website consisting of statically generated HTML pages. It provides minimum interactivity with the rest of the system and was therefore omitted from the privacy threat model DFDs.

■ Figure 4.1 Context model DFD of the CTU Survey



■ Figure 4.2 Level 1 DFD of the CTU Survey



is therefore focused less on the internal operations of the system and it is hence not required to decompose the internal processes further. [3] The level of detail which can be obtained from the context model DFD and the level 1 DFD was deemed sufficient for the privacy threat analysis.



■ **Table 4.1** Mapping privacy threat categories to DFD elements [14]

Privacy threat category	Actor	Data flow	Data store	Process
Linkability	X	X	X	X
Identifiability	X	X	X	X
Non-repudiation		X	X	X
Detectability		X	X	X
Disclosure of information		X	X	X
Unawareness	X			
Non-compliance		X	X	X

## 4.2 Mapping threats to DFD elements

In the second step of the analysis the elements depicted in the system DFDs are mapped to the LINDDUN threat categories. Different DFD element types might be more or less susceptible to certain privacy threat categories. For example, the Unawareness threat category is relevant only for external actors (users) who might be unaware of the consequences of providing their personal information to the system. The specific threats from this category should therefore be examined in relation to the DFD elements of type actor. [3] I included the template provided by the authors of LINDDUN indicating which threat categories are relevant to each DFD element type in table 4.1.

The mapping template 4.1 served me as a base for creating the mapping table for the CTU Survey system, which was meant to be used as a checklist throughout the analysis. When creating the mapping table customized for the specific system, each 'X', which represents a potential threat posed to a DFD element, must be either documented by at least one threat or an assumption should be made and explicitly written down to explain why the DFD element is not susceptible to the given threat. [3]

## 4.3 Threat elicitation and documentation

The threat elicitation step is the core step of the LINDDUN methodology. In this step the intersections between DFD elements and threat categories from table 4.2 are examined to identify privacy threats relevant to the CTU Teaching Survey system. The LINDDUN framework provides a knowledge base and a catalogue of threat trees, each of which corresponds to one of the crossed intersections in table 4.1, to support the threat elicitation process. An example of a privacy threat tree from the LINDDUN catalogue is depicted in figure 4.3.

The identified threats are documented as either threat scenarios or misuse cases (MUC). MUCs can be considered as use cases from the misactor's point of view. [3] Each misuse case description consists of the title, summary, primary misactor (attacker), basic path, and consequences. The OWASP Threat Dragon modeling tool [17] provides a template to document threats for each DFD element.

### 4.3.1 General assumptions

The '-' characters in the mapping table 4.2 refer to threat types which would be generally considered relevant for the given DFD element type, but I made the decision, after careful consideration, to not take them into account in relation to the CTU Survey DFD 4.2. Also, even

■ **Table 4.2** Mapping privacy threats to the CTU Survey DFD elements

Element type	Threat Target	L	I	N	D	D	U	N
Data store	database	X	X	X	X	X		X
Data flow	user requests (user-frontend)	X	X	X	X	X		X
	requests (frontend-backend)							
	responses (backend-frontend)	X	X	X	X	X		X
	survey responses (frontend-user)							
	user redirect (IdP-frontend)	-	-	-	-	X		X
	SAML assertion (frontend-IdP)	-	-	-	-	X		X
	user credentials (frontend-LDAP)	-	-	-	-	X		X
	verification message (LDAP-frontend)	-	-	-	-	X		X
	GA cookies (frontend-GA)	X	X	X	X	X		X
	JDBC query (backend-database)	-	-	-	-	-		X
data (database-backend)	-	-	-	-	-		X	
Process	frontend	X	X	X	X	X		X
	backend	X	X	X	X	X		X
Actor	user	X	X				X	

though in theory all 'X' from the mapping table should be examined individually for privacy threats, in practice the authors of LINDDUN advise to apply the technique of *reduction*, to combine several 'X's which are applicable to the same threat. Reduction can be done either for 'X's that involve DFD elements of the same type (e.g. all data flows), or when the threats, which involve the same type of data (credentials, or anonymous data, non-sensitive data, etc.) and result in the same consequences, affect multiple DFD elements. [3]

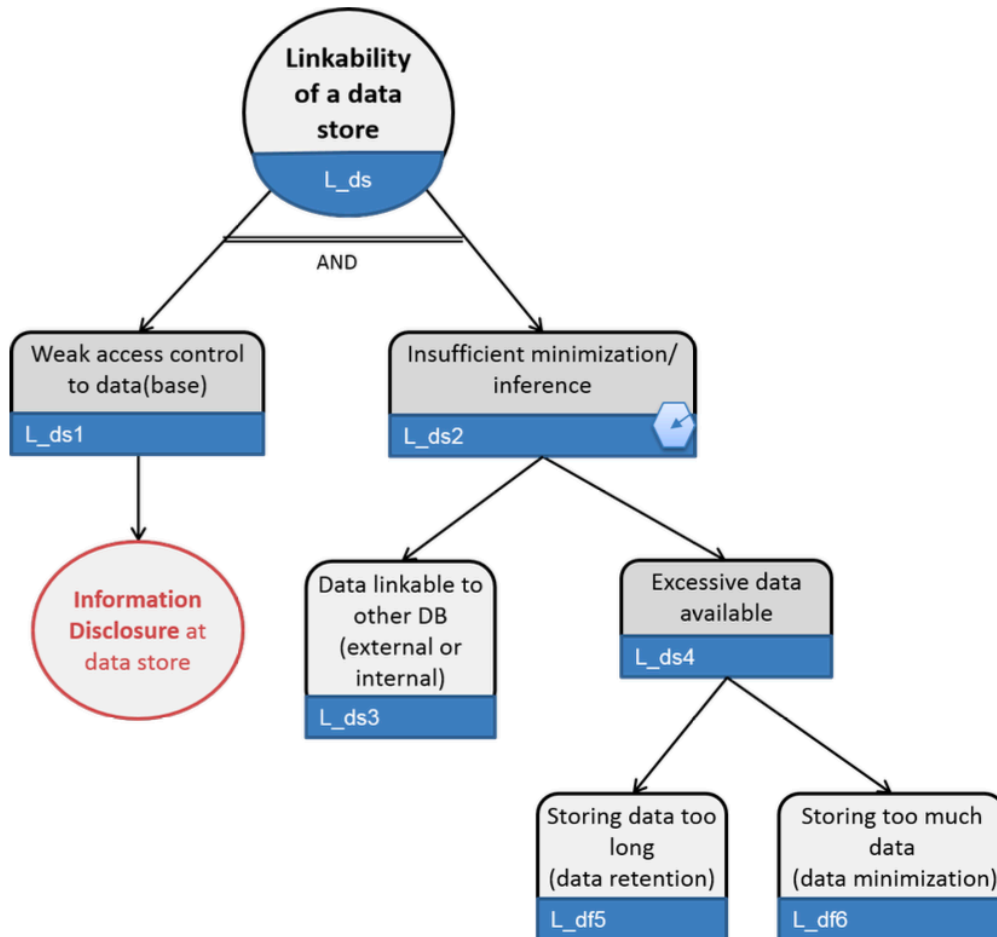
All assumptions of that kind need to be documented during the analysis process. [3] Assumptions also include choices to trust an element of the system to behave as expected. This section provides the reasoning behind each of the assumptions made about the threat targets and the corresponding LINDDUN threat categories.

**User-Survey data flows and internal processes** Reduction is applied on the data flows between the user and the application, data flows between the application front-end and back-end, as well as the internal processes. All of these elements in conjunction process same type of data. Furthermore, privacy threats of separate processes are very rare and can usually be determined as not applicable to the system. [20, 3] Therefore, the system as whole and the external data flows are examined for privacy threats rather than examining the frontend and backend processes separately.

**Data flows between backend and database** The internal data flows between the backend and the database are omitted from the privacy threat analysis. The focus of a privacy threat analysis is mostly on the data flows between the application and external entities. However, the database itself is not excluded from the analysis since it can be accessed by other means (by the application or database administrator, developers and testers).

**Authentication services** The data flows between the external authentication servers and the CTU Survey contain user credentials and need to be secured from unauthorized access. These

## Linkability of a data store



■ **Figure 4.3** Linkability of a data store threat tree from [20].

data flows (especially outgoing data flows implemented within the application) will be examined for possible misconfiguration and disclosure of information. All methods of authentication (SSO, LDAP, internal login, JWT) will be examined during the source code and vulnerability assessment phase.

**Unawareness of actors** Unawareness is only applicable to the actor elements since only an external entity (a subject) can be (un)aware of the consequences of sharing their personal information with the system. [3] In case of the CTU Survey, the only actor considered within the scope of the analysis is the user.

**Non-compliance applicable to the whole system** The non-compliance category is applicable to the system as a whole, since the whole system and all of its internal components together need to be configured and work in way which ensures privacy policies, legislative rules and subject's consents. [1]

■ **Table 4.3** DREAD risk rating table [22]

Risk rating	Result
High	12–15
Medium	8–11
Low	5–7

## 4.4 Prioritizing threats

The threats, which were elicited during the previous step of the analysis, must be prioritized before the proposal of suitable mitigation strategies. In general, risk is calculated as a function of the likelihood of the MUC and its impact. [3]

$$\textit{Risk} = \textit{likelihood} * \textit{impact}$$

The LINDDUN methodology does not include any specific privacy-risk assessment technique. Some of the recommended risk assessment techniques include the OWASP’s Risk Rating Methodology [21] or Microsoft’s DREAD [22]. I use the DREAD model for risk assessment.

The acronym DREAD stands for Damage potential (impact on assets), Reproducibility (how easy it is to reproduce the attack), Exploitability (how easy it is to launch the attack), Affected users (how many users would be affected by the attack) and Discoverability (how easy it is to find the vulnerability). Each threat is assigned rating values from 1–3 for every item in the DREAD acronym. Final score is the sum of the rating values. The severity is defined as low, medium or high based on table 4.3. I included an excerpt from the DREAD risk evaluation table in table B.1.

The Threat Dragon includes the option to assign *Priority* to each documented MUC as either Low, Medium or High. Since this scale corresponds to the DREAD final rating scale, I used this feature to document the risk value of each MUC.

I furthermore re-evaluated the resulting risk for privacy threats based on the level of risk they impose on the survey respondents anonymity. This meant to order the threats based on the LINDDUN categories from having the most to the least direct impact on students’ anonymity. The resulting order of threat categories with regards to the impact they have on anonymity is: Identifiability, Linkability, Disclosure of Information, Detectability, Non-repudiation, Unawareness and Non-compliance. Identifiability is defined as the direct opposite of anonymity and Linkability can lead to Identifiability, whereas Unawareness and Non-compliance (even though they can have serious impact on user’s privacy) have the least direct influence on students’ anonymity.

I included both the DREAD risk evaluation table B.1 and the final results of the privacy-risk evaluation step in table B.2 in the Appendix B of the thesis.

### 4.4.1 Examples of privacy-risk assessment using DREAD

DREAD as a security risk assessment model has its advantages. Its application is rather straightforward and offers a level of flexibility (it is applicable to different kinds of systems and situations). [23] One of the disadvantages of using DREAD is that the scoring of each of the risk categories is more subject to the analyst’s interpretation. [24] In the following section I provide a few examples of my approach to the assessment of privacy threats using DREAD.<sup>2</sup>

<sup>2</sup>Another disadvantage, which should be taken into account when applying DREAD, is that the Discoverability category promotes security through obscurity, which creates a false sense of security and is discouraged to use as a security measure. [24]

■ **Table 4.4** Excerpt from the DREAD risk rating table B.1

Privacy threat	D	R	E	A	D	total	priority
Use of production data for testing purposes in publicly accessible test environments	2	3	2	3	2	12	high
JWT token misconfiguration	3	3	3	3	2	14	high
Missing consent cookie banner	1	3	1	3	3	11	medium
Linkability of survey responses	2	2	2	2	2	10	medium
Default settings might lead respondents to disclose more data than they wished to disclose	1	3	2	2	3	11	medium
Students are not aware of the consequences of disclosing information about themselves	2	2	1	2	3	10	medium
Survey respondents are unaware of stored data	2	3	1	2	1	9	medium
Too much data stored in the database leads to linkability	1	2	2	2	1	8	medium
Redundant data stored with each survey response	2	3	1	2	1	9	medium

■ **Table 4.5** Excerpt from the privacy-risk evaluation table B.2

Rank	Privacy threat	priority	LINDDUN category
1	Weak implementation of SAML assertion	15	Disclosure of information
2	Weak implementation of SAR security	15	Disclosure of information
3	Disclosure of user credentials within a LDAP request	15	Disclosure of information
4	Disclosure of information within a LDAP verification message	15	Disclosure of information
5	JWT token misconfiguration	14	Disclosure of information
6	Unencrypted communication in test env.	14	Disclosure of information
7	Publicly accessible STG and DEV env. lead to information disclosure	13	Disclosure of information
8	Student's evaluation of a specific course detectable in STG env.	12	Detectability
9	Use of production data for testing purposes in publicly accessible test environments	12	Non-compliance

**JWT token misconfiguration** The CTU Survey uses JSON Web Tokens (JWT) for HTTP requests authentication and authorization. The application relies on the correct implementation of JWT to ensure access control. Improper implementation or misconfiguration of the authentication token would allow an attacker to gain access to resources which they should not have access to and to take unauthorized actions on those resources. I assessed the damage potential of this threat as high, since it has the potential to compromise the confidentiality and integrity of the whole system. Once the attacker discovered the vulnerability, it would be fairly easy to reproduce the attack, therefore I assessed the value of reproducibility also as high, as well as exploitability and affected users (the exploitation of this vulnerability would affect all users of the system). I assessed the discoverability property as medium. It would probably (depending on the actual manifestation of this threat in the system) require an attacker with advanced technical knowledge to discover the error. The overall risk score of this threat is 14, which is high priority.

**Too much data stored in the database leads to linkability** This threat is represented as a branch in the *Linkability of a data store* threat tree, which can be found in the “*LIND(D)UN threat tree catalog*” [20].

The insufficient minimization of data leads to linkability which might lead to inference or identifiability. A requirement for linkability in the database is weak access control. Linkability in the data store becomes a privacy threat in a system which has a weak access control to the data store. [20] Considering the restricted and secured access to the CTU Survey database, I assessed the damage potential of this threat, as well as the discoverability, as low. The attacker would have to have access to the data store to take advantage of the information concerning the potentially linkable data.

It would have to be either a malicious insider with access to the PROD, DEV or STG database, or a skilled (persistent) attacker, who would be able to gain access to the database by other means. I assessed the reproducibility and exploitability, as well as affected users, as medium. The disclosure of information leading to linkability at the data store would affect all users of the system whose data are present in the database. However, the linkability threat mostly affects the Survey respondents and their anonymous responses. The overall risk score of this threat is 8, which is medium priority.

#### 4.4.2 Next phase after threat evaluation

The LINDDUN methodology adheres to the *privacy by design* paradigm to introduce privacy early on in the development lifecycle. The authors of LINDDUN encourage to incorporate LINDDUN into the SDL and emphasize that it can also be applied to existing software systems (such as the CTU Survey). [3]

The next step of the LINDDUN methodology after the elicitation and evaluation of threats is the determination of suitable mitigation strategies. This applies to a software system under development. In case of an existing application the next step after threat modeling would be to take a closer look at the system and determine whether any of the previously elicited threats are actually present in the system in the form of vulnerabilities.

I used the list of threats from table B.2 and examined the CTU Survey source code and database and the data flows between the application and external entities for manifestations of these threats. The following chapter presents the outcomes of the source code examination.

# Source Code Examination and Vulnerability Assessment

*The focus of the following chapter is exploring the source code, database, and general workflow of the application, as well as vulnerabilities and privacy issues discovered during the analysis. The source code repositories of the backend and frontend modules are not publicly available, and access to all versions of the database (PROD, STG, and DEV) is also restricted. For the purpose of the anonymity analysis, I have been granted developer access to the source code repositories by Ing. Michal Valenta, Ph.D., as well as access to the DEV and STG versions of the Survey database.*

*The examination focuses on the privacy threats from table B.2 from the highest level of priority, beginning with an examination of the user authentication and authorization methods used within the application, followed by the examination of the anonymization process of the survey responses.*

## 5.1 User authentication

This section describes the user login and user authentication mechanisms implemented within the CTU Survey. I examined the implementation of these mechanisms in the application source code as well as the communication between the application and the authentication services.

After analyzing the HTTP requests and the application source code, I have not come across any evidence that suggests a misconfiguration that could result in information disclosure.

The CTU Survey offers three mechanisms for users to log into the application. Single Sign-On (SSO) is the primary login method offered to all internal teachers and students with a valid CTU identity. LDAP and internal login are secondary login methods. The LDAP option was added for students who have already finished their studies and do not possess a valid CTU identity anymore. In reality, any student can use LDAP to log in. The third option is the internal login, which was created for external teachers who do not possess a valid CTU identity.

The privacy threat analysis is mostly focused on threats related to the data flowing from the application to external entities. [3] Both the SSO and the LDAP are external authentication services provided by the CTU CIC [6]. Even though the SSO and LDAP services themselves are out of scope of the analysis, I examined their data flows with the CTU Survey to ensure the secure handling of user credentials.

### 5.1.1 SSO

Single-sign on is a session and user authentication service that permits a user to use one set of credentials to access multiple applications. [25].

The implementation of SSO used by the CTU is Shibboleth [6]. It is a web-based SSO system usually made-up of the following components: the Identity Provider (IdP) and the Service Provider (SP). The IdP is located at the home organization (which is CTU in case of the CTU Survey) and is responsible for user authentication and providing user information to the SP. The SP is responsible for protecting an online resource, accepting information from the IdP and ensuring user authorization. [26]

The CTU IdP is located on <https://idp2.civ.cvut.cz/>. The Anketa CTU uses an external `mod_shib` module configured on an `httpd` apache server as its SP. [8]

The general login flow using Shibboleth can be described as follows: user chooses the SSO login from the CTU Survey login page. The Survey SP returns a service authentication request (SAR) back to the user's browser, which redirects the user to the login site of the IdP, where the user can insert their credentials. After successful authentication, the IdP generates a response with a SAML assertion and returns that to the browser. A SAML assertion is a cryptographically signed XML document which contains information about the user. The browser forwards the SAML assertion document to the SP. SP validates the signature of the SAML assertion (PK). The SP creates a session for the user, based on the information in the assertion received from the IdP. SP returns the protected resource to the browser based on what the user is allowed to access.

If the Shibboleth user authentication completes successfully the CTU Survey backend generates the JWT to authorize the user. Each subsequent request sent between the frontend running in the user's browser and the backend contains both the `__shibsession__` session cookie as well as the JWT.

I examined the HTTP requests with the SAR generated by the SP and the SAML assertion returned from the IdP. I tested that the communication between the IdP and the browser is encrypted, the SAR does not contain any information about the user, the SAML2 assertion is encrypted so it can be decrypted only by the private keys held by the SP and that the `__shibsession__` cookie is tagged with the `HttpOnly` tag to prevent client-side scripts from accessing its data. [27, 28]

### 5.1.2 LDAP

*“The Lightweight Directory Access Protocol (LDAP) defines a way for clients to send requests and receive responses from directory services.”* [29] The LDAP used to access an active directory is (similarly to the Shibboleth) an authentication service operated by the CTU CIC. The server is located on `ldaps://ldap.cvut.cz:636`. The workflow of the login via LDAP could be described as follows: user inserts their username and password, which are sent in an HTTP request to the CTU Survey backend. The backend uses the credentials provided by the user to authenticate the user against the LDAP server. This is done via JNDI API [29]. The backend tests the connection to the LDAP server. If the connection fails, the backend throws a `LoginException` and terminates the login process. If the connection succeeds the backend generates a new unique JWT to authorize the user and returns it to the client. That concludes the authentication process.

### 5.1.3 Internal login

The internal login was added later to the application for the convenience of external CTU instructors who do not possess the CTU identity. It is implemented within the application and does not use any third party service for user authentication. The user credentials are stored in the



survey database. The passwords are hashed together with a salt generated randomly for each user.

When adding a new user to the system, the user is provided with a one-time token to create their password. This token had been previously generated and stored into the survey database together with the user's id. The user can set their new CTU Survey password using the URL: <https://anketa.is.cvut.cz/html/anketa/set-password>. After the user submits their password, the frontend forwards the HTTP request containing the user's one-time token together with their password to the Survey backend. There the request is parsed, the backend validates that the one-time token provided within the request corresponds to one of the tokens present in the survey database. The token is then deleted from the database. A hash of the provided password is created together with a randomly generated salt and persisted together with the user's id in the database. If a user forgets their password, their previous record will need to be deleted from the database. The user will then need to repeat the process of obtaining a new one-time token and using it to create a new password.

The conclusion from my examination of the functionality of creating new user passwords was that from a functional point of view the feature was not sufficiently tested. The frontend either does not give any indication, or indicates success in case of a failed attempt to create a new password.

There is also a weak password policy in place for the external teachers' passwords. The application demands passwords to be at least six characters in length, which is insufficient. Passwords chosen and memorized by the user should be at least 8 characters in length. Passwords that are too short are vulnerable to brute force attacks as well as dictionary attacks. [30]

## 5.2 Publicly available DEV and STG environments

There were several listed privacy threats of high and medium priority associated with the existence and insecure set-up of the publicly accessible DEV and STG environments. These environments were exposed on the internet, did not use encryption to secure communication, contained user production data and gave anyone the option to log in and view any user account without the necessity to provide valid credentials (it was sufficient to insert the user's CTU username or ID in case of the external teachers). In this section I address each of the threats (or groups of threats with common source) applicable to the deployment environments and discuss the potential for exploitation and privacy violations resulting from these threats.

### 5.2.1 Production data in publicly accessible environments

The use of user production data in testing environments may violate GDPR regulations, even if access to the testing environments is restricted. This is particularly true if the user data is personal and if the users did not provide a specific consent for the use of their data for testing purposes. [31] In the case of the CTU Survey, the access to the DEV and STG environments was not restricted and users were unaware of the use of their data as well as the existence and insecure configuration of the deployment environments.

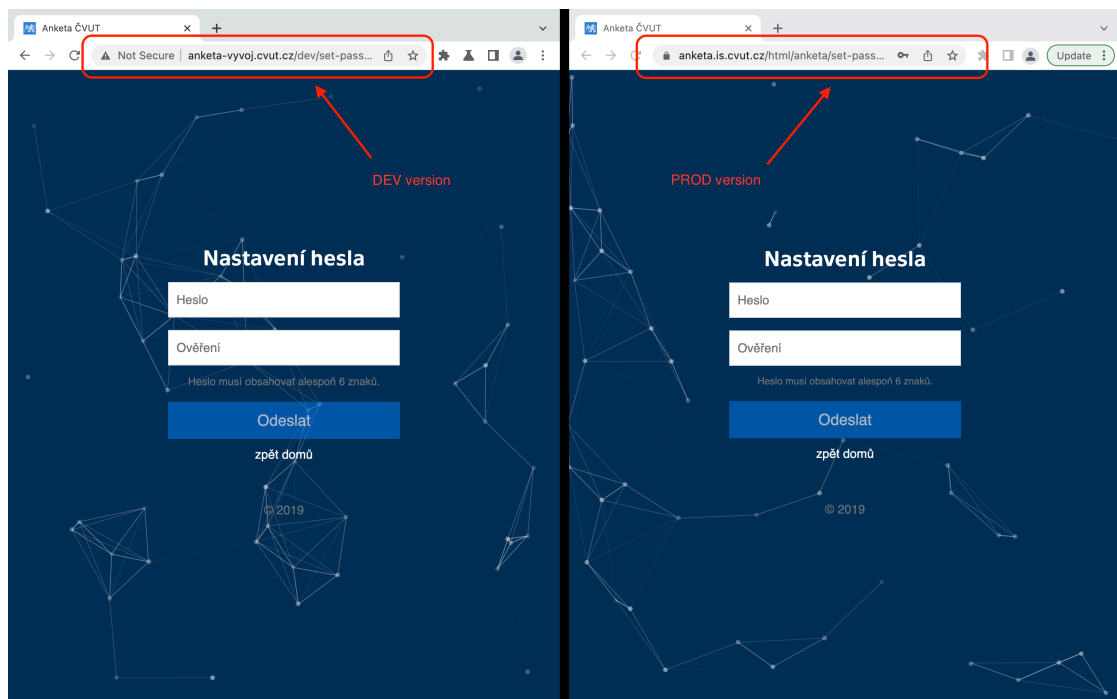
The DEV and STG environments allowed to log in as any user (any CTU student or teacher) and view the content on the user's Survey account. This threat could be categorized as information disclosure since it would allow the attacker to gain unauthorized access to the CTU Survey users' data without their knowledge and consent.

Since the attacker would have access to the same data as the users (with one week delay in case of the STG environment), this threat could have various implications considering the students' anonymity and overall users' privacy.

It would, for example, allow a student to view the content on any CTU instructor's account. This could give the student an advantage and (depending on the current phase of the survey

instance) the ability to view results of the current Survey even before their official publication. This threat could be categorized as non-compliance since it does not comply with the official rules and regulations of the CTU Survey system. [6]

There is a feature present in all application environments of the displaying module which allows students to filter and view Survey results only for courses for which they submitted their survey answers. The misactor, who could be a CTU student or teacher, may be motivated to link responses from a specific course the individual student's identity. To achieve this, they can log into the STG environment as a student and abuse the filter functionality. This would lead to detectability. The misactor would be able to detect whether a student submitted their responses for a specific university course or not.



■ **Figure 5.1** Password setting page — DEV environment and PROD environment

These individual examples of possible MUCs of the publicly available DEV and STG versions highlight the overarching issue which is the abundance of data which the misactor would potentially have access to and which they could abuse without the victims' knowledge (the victims in this case would potentially include all CTU students and teachers). The misactor gaining access to the DEV and STG versions of the CTU Survey would be presented with virtually all data of all of the application users and would be able to link the data and abuse the knowledge in various ways.

## 5.2.2 Use of HTTP for communication

The DEV and STG environments (unlike the PROD version) used the insecure HTTP protocol for data transfer.

One example of how an attacker might be able to abuse the insecure set-up of the DEV environment would be the possibility of successfully launching man-in-the-middle (MITM) attacks to steal user credentials, admin credentials from the administrator module (which is also publicly accessible and uses HTTP) or alternatively to obtain the one-time token generated for setting an external teacher's password. In this scenario the attacker (who could be a malicious outsider

or an insider) would be motivated to steal the victim's token or trick them into providing their valid credentials to gain access to their production Survey account. They could take advantage of the fact that the DEV and PROD versions of the password setting page look identical (except for the URLs and the security icon located next to the URL, indicating that the communication in the DEV version is not secure, which can be seen in figure 5.1).

The attacker then attempts to trick the victim into clicking on the link to the password setting page of the insecure DEV version instead of the valid PROD version. The victim tries to set their password in the DEV environment. After the victim submits the web form, the attacker intercepts the communication between the victim's browser and application backend, steals the victim's credentials, and uses it to log in or to set the victim's password in the PROD environment.

The successful execution of such an attack could have a negative impact not only on the CTU Survey system but also on any other CTU system that uses the CTU username and password for login credentials, including any system that uses the SSO Shibboleth for user authentication.

## 5.3 JWT token forgery

The following section describes the process of uncovering and mitigating a critical security vulnerability discovered during the source code examination phase. This vulnerability can be classified as a security misconfiguration. It occurred as a result of the existence and configuration of the DEV and STG environments, as well as the internal implementation of user authentication and authorisation using the JSON Web Tokens (JWT).

### 5.3.1 JSON Web Token

The CTU Survey uses JWT for communication between the client and server. JWTs are meant to provide user authentication and authorization. By utilizing JWT, the system can ensure that each individual user is granted access only to the content that is tailored to them and prevent users from taking unauthorized actions.

JWT is a compact and self-contained way for secure transfer of information between parties as a JSON object. This information is digitally signed and can be therefore verified and trusted. The JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA. [32]

JWTs consist of three parts separated by dots. The header part typically (but not always) contains information about the type of the token, which is JWT, and the signing algorithm being used. The second part of the token is the payload, which contains information about the user and additional data. [32] In the case of the JWTs generated by the CTU Survey, the payload contains a username, a time of issuing and a time of expiration. An example of a JWT header and payload generated by the CTU Survey backend can be seen in figures 5.2 and 5.3. Both the header and payload parts are Base64Url encoded.

■ **Figure 5.2** JWT header

```
{
  "alg": "HS256"    // signing algorithm
}
```

The third part of JWT is the signature. The encoded header and payload are signed together using a secret and the specified signing algorithm (figure 5.4). The signature is used to verify the integrity of the information being transferred.

■ **Figure 5.3** JWT payload

```
{
  "sub": "USERNAME",      // CTU username
  "iat": 1678996132,     // issued at
  "exp": 1679600932     // expiration time
}
```

■ **Figure 5.4** JWT signature created by signing the encoded header and payload together with the secret [32].

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

The final output are three Base64Url encoded strings separated by dots (figure 5.5). In this compact form the token can be easily passed in HTML and HTTP environments. [32]

■ **Figure 5.5** Example of a JWT [32]

```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJVU0VSTk
FNRSIsIm1hdCI6MTY3ODk5NjEzIiwiaWF0IjoxNj
jc5NjAwOTMyfQ.mfHmpLcHW4iSvGUDT3RJ1S3c7
L-wtnJYTZ1f0nrBiiA
```

### 5.3.2 CTU Survey JWT misconfiguration

In the case of the CTU Survey, the JWT is generated for each user after their successful login to the application.

User logs into the application through one of the available means (SSO, LDAP or internal login). Once the authentication phase is successfully completed, the backend generates a unique JWT, setting the username, current time and current time + expiration period as the *sub*, *iat* and *exp* JWT claims. The expiration period of each JWT is set to one week. The token is then signed with a secret, which is hard coded as a string of text in the *application.properties* configuration file. The newly generated unique token is then sent back to the client. It is then included in the Authorization header with each subsequent request from the client to the server.

The JWT authentication is stateless, meaning the server does not store any information about the already issued tokens. The only way to invalidate an existing token before its expiration date would be by changing the JWT secret.

The previously described way of issuing and using JWTs for authentication and authorization is standard and by itself does not impose any security risk to the users or the application. The security issue stemmed from the fact that all CTU Survey environments were configured to use JWTs for authentication and used the same secret to generate tokens. This allowed a potential attacker to generate JWTs for any user in the publicly accessible DEV or STG environments and then use them to authenticate as that user in the PROD version, without providing valid

credentials. This was a serious issue which could potentially compromise the integrity and confidentiality of the system.

The mitigation strategy suggested by my supervisor was to add another secret for generating tokens to the PROD environment. This way the vulnerability would be no longer present since the JWTs generated in the DEV and STG versions would not be accepted by the PROD backend and any attempt to forge the user token would result in an error. Adding the new secret to the PROD version would also immediately invalidate all tokens which were issued in the past week.

This solution was implemented by the CTU Survey development team within a few days after disclosing the vulnerability and is no longer present in the application (as of April 2023). I suggest to incorporate the test for this particular vulnerability into the application regression testing suite to prevent it from occurring again in the future.

### 5.3.3 PoC exploit

This subsection presents the proof of concept (PoC) exploit to demonstrate one possible scenario of how an attacker could abuse the JWT misconfiguration. The proxy feature of the Burp suite security testing tool<sup>1</sup> was used to examine and tamper with the content of HTTP requests which are sent in communication between the application client and the server.

In this scenario the attacker is aware of the existence of the publicly available STG and DEV versions. To discover the vulnerability at that point would be fairly easy. The attacker could simply try to replace their authentication token in the PROD version with a token generated in the DEV environment. There would be multiple possibilities of how an attacker could abuse the error. Depending on the phase of the current survey instance, users would be presented with different options within the application. E.g. during the second phase of the survey lifecycle, which is the time when survey opens for students, an attacker would be able to log in as any student and fill in their survey questionnaires.

During the survey evaluation phase an attacker could login to any course instructor account and tamper with their comments to survey results. This case is demonstrated in figures 5.6, 5.7, 5.8 and 5.9.

The first step would be to log into the DEV or STG environments using the victim's username. The backend generates a JWT which can be found and copied from the request's Authorization header.

The attacker then stores the victim's token and logs into the PROD version using a valid set of credentials. With their own token, the attacker is able to navigate to the results page of the CTU Survey and view comments which were written by the victim. At this point the attacker does not have the option to modify or delete the teacher's comments. However, after refreshing the page and replacing their tokens with the victim's tokens in all HTTP requests, the attacker can view the page as the victim and is able to modify or delete their comments.

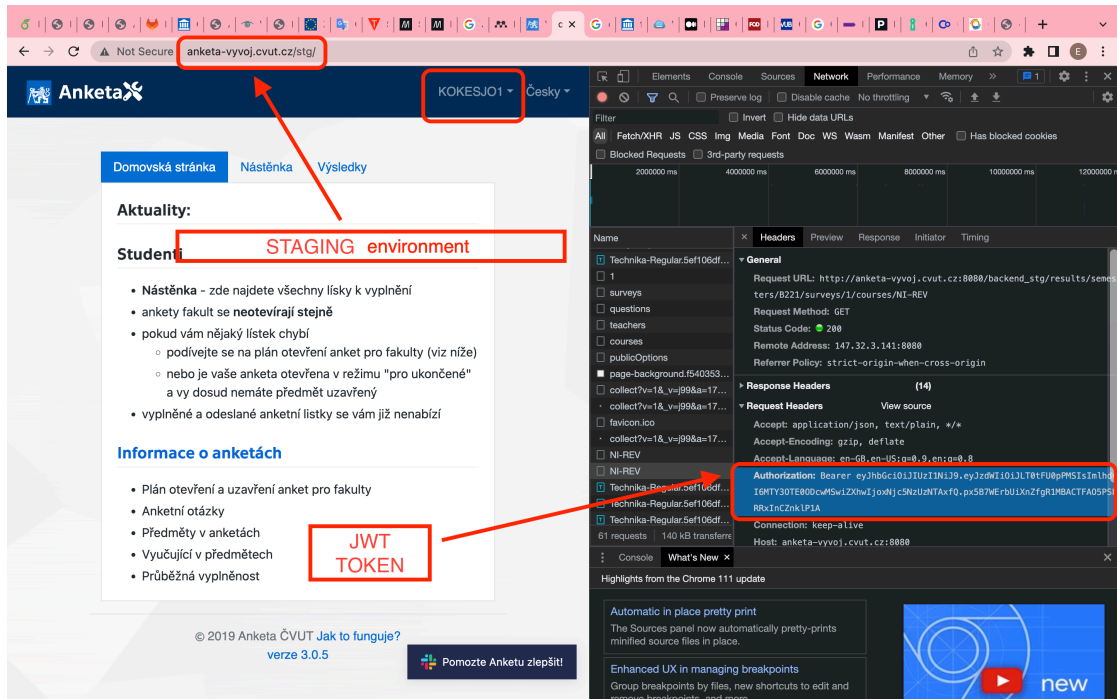
## 5.4 Separating student's identity from Survey responses

The process of separating the respondent's identity from their survey responses during submission can be considered the most critical process for the anonymity analysis. My aim in this section is to describe the workflow of the process and how the CTU Survey implements this central functionality to ensure the survey respondents' anonymity.

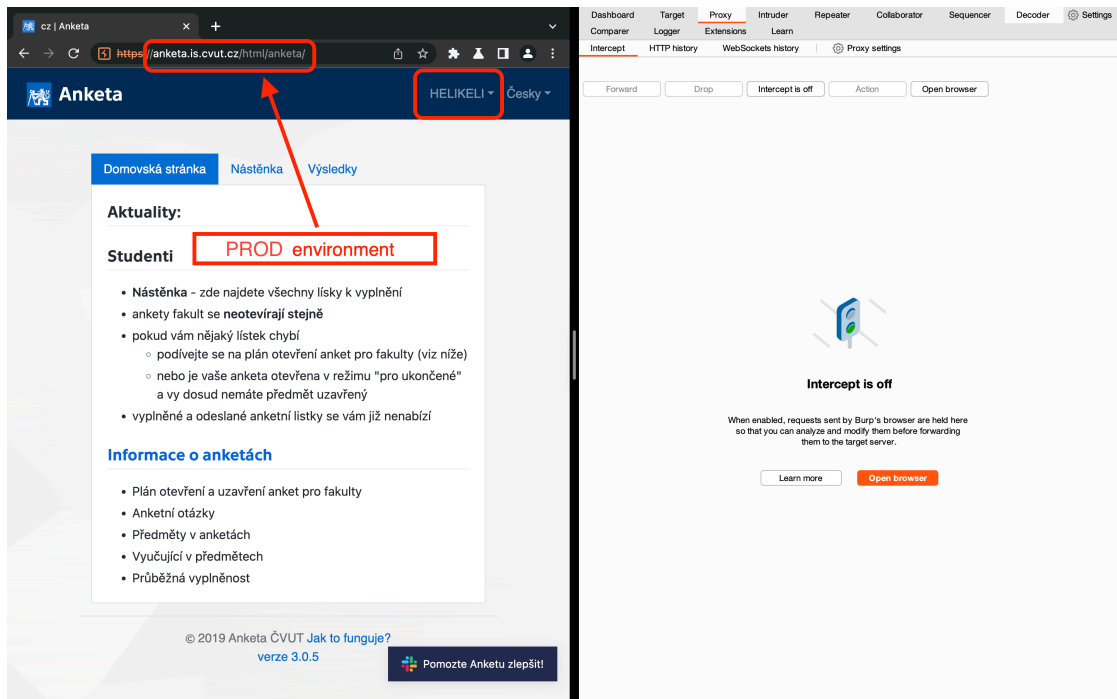
After the respondent fills in their survey response for a specific course and confirms the submission of their survey questionnaire, the application frontend first sends HTTP requests to the google-analytics.com 3rd party service which collects data about users' behavior on the application GUI. After that the actual request containing the survey response filled by the user

---

<sup>1</sup>Burp Suite Community Edition v2023.3.5, developed by PortSwigger Ltd.



■ Figure 5.6 JWT forgery: Generating teacher tokens in the STG environment



■ Figure 5.7 JWT forgery: Login into the PROD version as another user (student)

is sent to the application backend for processing. The communication between the application frontend and backend is secured via HTTPS.



payload and is included as a parameter in a query to search for the user in the Survey database. After the user is found, their username is included into the application security context (which is a feature provided within the Spring Boot Security library [33]) and can be later accessed from anywhere within the application.<sup>2</sup>

After the successful user authentication the backend processes the request body. First the application validates in the database that the student is eligible to submit responses for the given university course. The information about the submitted survey questionnaire is stored in the `COURSE_IN_SURVEY_FILLED_BY` table.

Then additional information about the student (excluding their student id and name) is saved to the `COURSE_EVALUATION` table. The additionally stored data consists of the following personal attributes: the student's grade in the course, their overall average grade, the code of their study program, the program type, the study form, the student branch code, the student year and the student course role. This information is stored into the database even if the student did not agree to add any additional information to their survey response. The separate survey response is then saved into the `COURSE_EVALUATION_ANSWER` table.

The examination of the process for anonymizing survey responses revealed privacy concerns with varying levels of risk both on the application frontend and within the data flows between the frontend and backend, as well as in the system database. My aim for the last two sections of this chapter is to address these issues and discuss their potential impact on the of survey respondents' anonymity.

## 5.5 User actions revealed within Sentry breadcrumbs

The CTU Survey uses the Sentry service to track errors and crashes. In case of an application exception being raised, an error report is sent to the `sentry.fit.cvut.cz` server and it can be later examined by the application developers.

The error report contains information such as username of the user whose session triggered the error, event id and timestamp of the event. It also includes a trail of the last one hundred events that happened prior to the crash. These Sentry events are called *breadcrumbs*. They are similar to traditional logs, but can record richer structured data. [34] In case of the CTU Survey these breadcrumbs contain a lot of information regarding the user's behavior on the web site. Each breadcrumb contains a timestamp, event category (e.g. a UI click event or a navigation event) and data or message which provides additional information regarding the user's behavior such as details about the clicked UI element.

Should an error occur shortly after a student submits a survey feedback, a trail of the user's behavior contained within the one hundred Sentry breadcrumbs sent to the Sentry server reveals information that describes exactly which survey responses for which courses were accessed, filled and submitted by the user together with the timestamp of each subsequent event. An example of the trail of Sentry breadcrumbs documenting the user action of filling a survey response is depicted in figure 5.10.

An internal misactor who would have access or would be able to gain access to the Sentry application reports as well as to the Survey database would be able to link the username, timestamps and additional information about the user's behavior to the anonymized survey responses stored in the application database and identify the respondent. Even without the access to the database, the information included within the Sentry breadcrumbs weakens the anonymity property since it reveals detailed information about the user's actions that are supposed to be anonymous.

---

<sup>2</sup>This process of user authentication with the JWT happens right before the processing of any HTTP request coming from the application client and is not specific to the submission of survey questionnaires.



## 5.6 Identifiability, Unawareness and Non-compliance

During the process of filling survey questionnaires, students are presented with the option to include additional information about themselves which will be later revealed in the application GUI together with their qualitative answers. The default setting of this feature does not comply with the privacy by default paradigm since it implements the opt-out principle by pre-selecting the respondent's study branch, year of study and role of the course attributes, leading the student to potentially disclose more data than they intended. The default setting for the survey questions should be fully anonymous. The default settings of this feature are depicted in figure 5.11.

This issue ties to another group of privacy threats which relate to the user's (un)awareness. The CTU Survey respondents might not be aware of the consequences of submitting their responses and disclosing additional information about themselves. The identifiability of a subject depends on the size of the anonymity set as well as on the number of known defining attributes which make the subject distinguishable from other subjects within the anonymity set. [1] In the case of the CTU Survey, the students might not know which combination of attributes will make their survey answers identifiable within the anonymity set of a specific university course.

Threats from the non-compliance category which were assessed as relevant in the context of the CTU Survey relate to the application's use of 3rd party services, namely the Google Analytics (GA) service, the Sentry error logging tool and the Slack feedback tool. The use of GA within the CTU Survey does not comply with the current privacy legislation since it does not inform the users and does not give them the option to restrict the use of analytical cookies during their first visit of the main page (this includes all potential users, not just the logged-in users). The issues with the Sentry error logging tool were already discussed in previous sections. Regarding the Slack feedback tool, users might not be aware that their feedback provided using this tool is not anonymous (unlike the survey responses implemented within the filling part of the module).

## 5.7 Insufficient data minimization in the database

The examination of the process of anonymization of survey responses revealed that there is a set of information about the responding student stored in the database with each survey response. The extra stored student attributes are redundant and do not serve any evident purpose within the application. They are also stored without the respondent's knowledge and although they do not contain the student's username or id, they contribute to the insufficient minimization of data in the database. There is an excessive amount of data stored in the application database and the data is stored for too long. This leads to insufficient minimization and can lead to linkability of data or possibly even re-identification of survey responses.

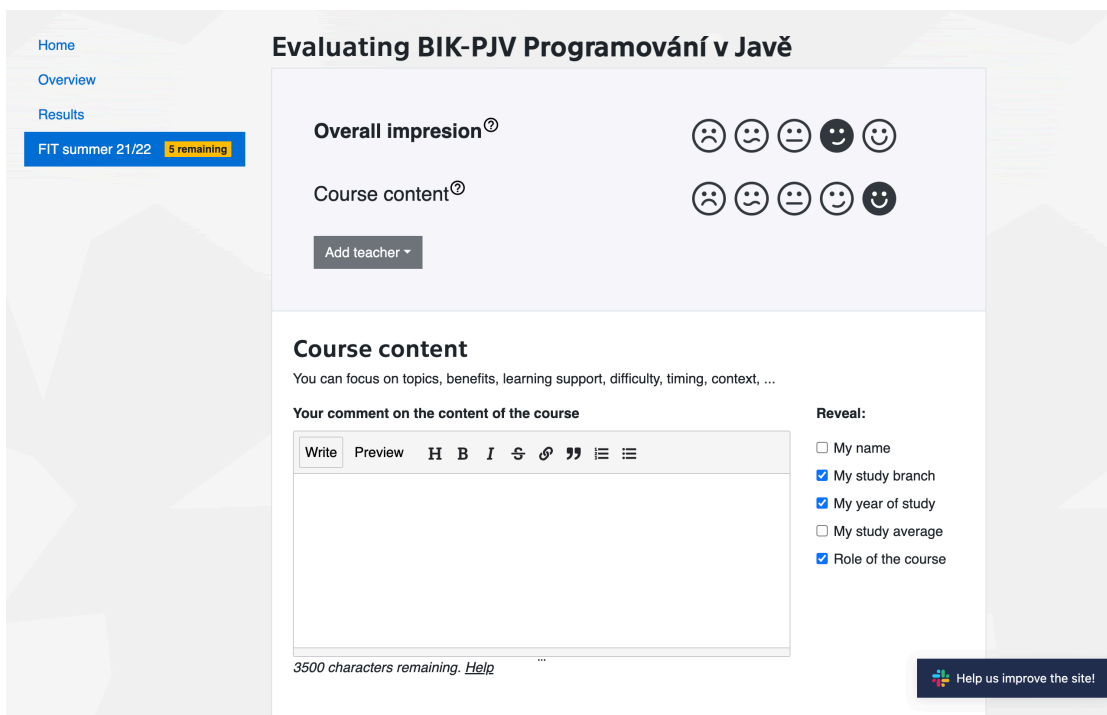
■ **Figure 5.10** Sentry error report containing trail of events revealing the user actions of navigating on the website, clicking on the quantitative feedback buttons and then filling the text-based questions for a specific course

```

    "user": {
      "username": "HELIKELI"
    },
    "breadcrumbs": [
      {
        "timestamp": 1679232486.194,
        "category": "xhr",
        "data": {
          "method": "GET",
          "url": "http://anketa-vyvoj.cvut.cz:8080/backend_dev/courses",
          "status_code": 200
        }
      },
      ...

      {
        "timestamp": 1679236567.313,
        "category": "navigation",
        "data": {
          "from": "/dev/surveys/1B212",
          "to": "/dev/surveys/1B212/ticket/1445506"
        }
      },
      {
        "timestamp": 1679236568.558,
        "category": "ui.click",
        "message": "div.col > div > div.row > div.col-xs.smiley-wrapper"
      },
      {
        "timestamp": 1679236569.137,
        "category": "ui.click",
        "message": "div > div.row > div.col-xs.smiley-wrapper > div.smiley > svg#Layer_1"
      },
      {
        "timestamp": 1679236569.97,
        "category": "ui.click",
        "message": "div > div.row > div.col-xs.smiley-wrapper > div.smiley > svg#Layer_1"
      },
      {
        "timestamp": 1679236572.03,
        "category": "ui.click",
        "message": "a.list-group-item.list-group-item-action.text-primary"
      },
      {
        "timestamp": 1679236572.036,
        "category": "navigation",
        "data": {
          "from": "/dev/surveys/1B212/ticket/1445506",
          "to": "/dev/surveys/1B212/ticket/1445506#question1"
        }
      }
    ]
  }

```



■ Figure 5.11 Default settings for the text-based survey questions.

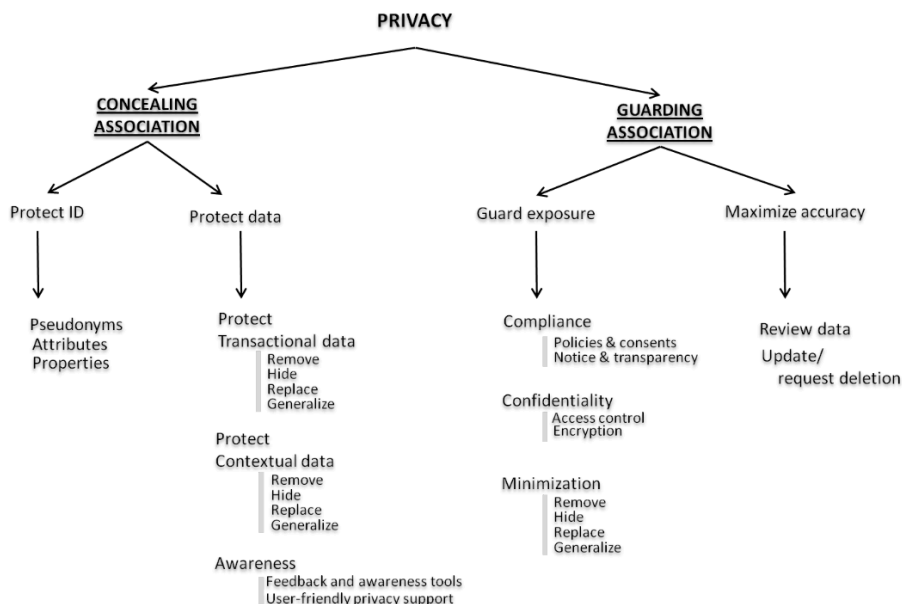


# Evaluation of Findings and Mitigation Strategies

The following chapter presents the final outcome of the privacy threat analysis, which is the resolution of threats and vulnerabilities discovered within the system in the form of suitable mitigation strategies. The chapter also includes additional suggestions for enhancing anonymity of the survey responses.

## 6.1 LINDDUN solution space

The last two steps of the original LINDDUN methodology, which are part of the LINDDUN solution space, are elicitation of mitigation strategies and selection of corresponding privacy enhancing techniques (PETs). [3]



■ Figure 6.1 Taxonomy of LINDDUN mitigation strategies [1]

The methodology presents two major strategies for obtaining privacy: the *proactive* approach, in which the associations between users and their transactions and personal information are controlled in order to ensure that users share as little information as necessary with the system, and the *reactive* approach, which deals with limiting damage by controlling and restricting the associations after they are disclosed to the system. [3] The taxonomy of LINDDUN mitigation strategies (which is depicted in figure 6.1) is not an exhaustive list of strategies but rather an overview of common mitigation approaches. [1]

### 6.1.1 Concealing association

The proactive approach (or concealment of associations) can be divided into two sub-strategies: (1) protecting the user’s identity during authentication and (2) protecting the data that is communicated to and throughout the system. [1]

The strategies proposed in the mitigation taxonomy for protecting the user’s identity include the use of pseudonyms or, to achieve even stronger anonymity, using properties such as anonymous credentials or other zero-knowledge proofs as authentication methods. [1]

The second branch of the *Concealing Association* sub-tree in figure 6.1 deals with strategies for protecting the data that are being communicated. There is a further distinction made between the strategies that concern core data protection and those that are meant to raise awareness about the sharing of information, and also between the strategies that protect transactional data and the strategies to protect contextual data (or the metadata related to the communication). The high-level strategies for data protection proposed by LINDDUN are (both for the transactional as well as the contextual data) to (1) remove (sensitive) information, (2) hide the data, (3) replace part of the information or (4) generalize it. [1]

The final data protection strategies address threats from the Unawareness category and are meant to make users more aware of the consequences of sharing data. The system can enhance the user awareness by implementing feedback and awareness tools, and by providing user-friendly privacy support to the user. [1]

### 6.1.2 Guarding association

The reactive approach (or guarding associations) strategies is also divided into two sub-trees: (1) guarding exposure and (2) maximizing accuracy.

Strategies for guarding exposure address threats from the Non-compliance and Disclosure of information categories and propose means for data minimization. The mitigation strategies of non-compliance related threats include obtaining data protection compliance related to consents and policies, and notice and transparency. For protecting confidentiality, the proposed security enhancing strategies are access control and encryption. Finally, data minimization strategies are the same as for the protection of transactional data (remove, hide, replace and generalize). [1]

The final strategy for guarding association is the maximalization of accuracy. It allows the subjects to inspect and correct their information by either allowing them an easy access to the collected data in order to review it (which relates to user awareness) or by extending the access right and allowing the subjects to request updates or even deletion of their data. [1]

## 6.2 Privacy enhancing strategies for the CTU Survey

The following section provides examples of suggested mitigation strategies and solutions specific to the privacy threats applicable to the CTU Survey. The text showcases examples of solutions for threats which were assessed as having high or medium priority. The full list of discovered threats and their corresponding mitigation strategies is included in the document “Threat model report for the CTU Teaching Survey Web Application” as one of the attachments of this thesis.

### 6.2.1 Enhancing respondent awareness

Threats addressed in this section include: (12) *Default settings might lead respondents to disclose more data than they wished to disclose*, (14) *Student identifiability in Anketa GUI*, (18) *Students are not aware of the consequences of disclosing information*. The solutions proposed in this section correspond to the Protect Data – Awareness branch of the *CONCEALING ASSOCIATION* subtree in the LINDDUN mitigation taxonomy from figure 6.1.

In his original outline of the CTU Survey 3.0, Knap proposed a new feature for the application user interface (UI) which included the detailed anonymity settings for survey questionnaires. This setting allowed the survey respondents to chose which information they wished to post together with their survey answers. [7] This feature is present in the application to this day and had already been discussed before in this work. However, in the original Knap’s proposition the feature also included a set of system notifications which would warn the student in case his choice of attributes made them identifiable among the anonymity set of students enrolled in the same university course. The figure 6.2 shows a high-fidelity wireframe of that feature. The original picture can be found in Knap’s master’s thesis [7]. This feature was included in the system design but was not fully implemented. During the source code examination phase traces of the proposed feature were discovered in the frontend source files in the form of a Java Script (JS) function prepared to display an anonymity warning to the survey respondent in case the condition of identifiability was satisfied. The JS code of that function is depicted in figure 6.3.

Hodnotili jste vyučujícího **Ing. Josefina Smyšlená, Ph.D.** negativně. Abychom mohli předmět zlepšit, potřebujeme další informace o tom, co bylo v předmětu špatně, a jak to můžeme do příštího roku napravit. Zkuste být co nejkonkrétnější, pomůže nám to.

**Jaké problémy se ve výuce objevily?**

Hodnotíte vyučujícího Ing. Josefina Smyšlená, Ph.D.

**Zveřejnit:**

- Mé jméno
- Můj obor studia
- Můj ročník studia
- Můj studijní průměr
- Roli předmětu v mém stud. plánu

Nalezen pouze 1 student v předmětu s ročníkem 4.

■ **Figure 6.2** Awareness notification: “Only 1 student from year 4 found in this course” [7]

The anonymity-notification feature was included in the UI testing suite of the Anketa CTU 3.0 to check user reaction. The four participants included in the testing were presented with one of the prepared test scenarios. They were asked to evaluate a course instructor in a negative way. Then they were presented with the anonymity-notification which warned them about their current profile being identifiable. [7]

The test subjects did not pay much attention to the default anonymity settings of the survey questionnaire. When presented with the anonymity-notification, the test subjects did not understand its meaning. [7]

It is important to note that in the real world such a scenario would have potentially serious impact on the actual survey respondent who is deciding whether it is safe to include an honest but rather negative feedback without the risk of being identified and held accountable.

Therefore, the suggestion is to revise and include this feature, which was originally proposed by David Knap, into the current version of the application to enhance user awareness as well as the respondents’ trust in the survey system.

■ **Figure 6.3** Awareness notification: JS code snippet taken from the frontend source files (repository path: `src/Components/TextRating.js`) showcases the function prepared to display an identifiability alert.

```
renderBadgeWarning = (checked, type) => {
  const threshold = 1;
  if (checked && this.props.user[type] <= threshold)
    return (
      <span
        className='anon-warning badge badge-danger'
        data-toggle='tooltip'
        data-placement='right'
        data-trigger='hover focus manual'
        title={localization(
          'surveys_answer_anonymity_warning',
          this.props.user[type]
        )}
      >
        !
      </span>
    );
};
```

Another suggestion is to keep the settings for the anonymity of the respondents profile (depicted in figure 5.11) fully anonymous by default and let the students decide to include additional information (opt-in).

In general, the responsibility of users' awareness rests mostly on the system which should be designed, implemented and configured to support its users by providing information as well as privacy-friendly settings and privacy-enhancing options. The system should aid its users in deciding on what data they will share. It is also not the responsibility of the user to make sure the data that is collected by the system is correct. [1] However, in the case of the CTU Survey, there are certain aspects related to user awareness and identifiability which the system cannot easily influence.

Respondents should be aware of the information they provide within the qualitative text-based answers since the content provided in these answers can make them identifiable. For example certain language features can indicate an individual's membership in a specific minority group. E.g., since the vast majority (87 %) of the FIT CTU students are male [35] and the Czech language distinguishes between the feminine and masculine gender in the first person, some female students reported avoiding using the feminine gender when providing qualitative feedback since the information would make their responses identifiable. In such cases the system is not able to provide support and it is the student's responsibility to be cautious and consider possible consequences of sharing too much information or specific kinds of information with the system.

## 6.2.2 Restricting access to deployment environments

Threats addressed in this section include: (7) *Publicly accessible STG and DEV env. lead to information disclosure*, (9) *Student's evaluation of a specific course detectable in STG env.*, (10) *Use of production data for testing purposes in publicly accessible test environments*, (23) *Survey results disclosed before intended date*, (27) *Users are unaware that their survey accounts are*



*publicly accessible*

The threats associated with the insecure and privacy non-compliant set-up of the DEV and STG deployment environments have already been discussed in previous chapters. The proposed solutions for mitigating the most serious security and privacy threats include: restricting access to the environments (e.g. by allowing only a selected group of subjects who possess a valid set of credentials to access the environment) in combination with the use of a secure encrypted TLS protocol for data transfer.

To ensure that users' data is being handled with care and in compliance with the privacy regulations, my suggestion is to avoid using the production data of real application users for testing purposes.

Finally, it is important to ensure that users are aware of the use of their data.

### 6.2.3 Ensuring GDPR and other privacy-related compliance

Threats addressed in this section include: *(8) Application logging enables detection of survey responses, (13) Missing consent banner for non-technical cookies.*

The CTU Survey use of 3rd party services does not comply with the privacy legislation. The CTU Survey uses the Google Analytics (GA) 3rd party service to gather information about their users' behavior on the website. According to the General Data Protection Regulation (GDPR), as well as the Czech Office for Personal Data Protection, it is obligatory to inform users about the use of cookies in a form of a so called 'cookie banner' and to give users the option to choose which kinds of cookies they are comfortable with and which they would rather restrict. Without the user's explicit consent, the application is allowed to use only the technical cookies. [36]

It is necessary to configure the system to be privacy-friendly by default and to comply with the privacy legislation. This means to either add the cookie banner and ensure that it is implemented according to the legislation and to not use other than technical cookies by default, or alternatively to avoid using the GA all together. The Sentry error logging tool should be configured to not disclose the username and other details about the user's behavior that would make their actions of filling survey responses detectable and weaken anonymity.

### 6.2.4 Minimizing data stored in the database

Threats addressed in this section include: *(18) Re-identification of survey respondents, (19) Data identified by username, (27) Redundant data stored with each survey response, (30) Too much data stored in database leads to linkability, (31) Storing data too long leads to linkability.*

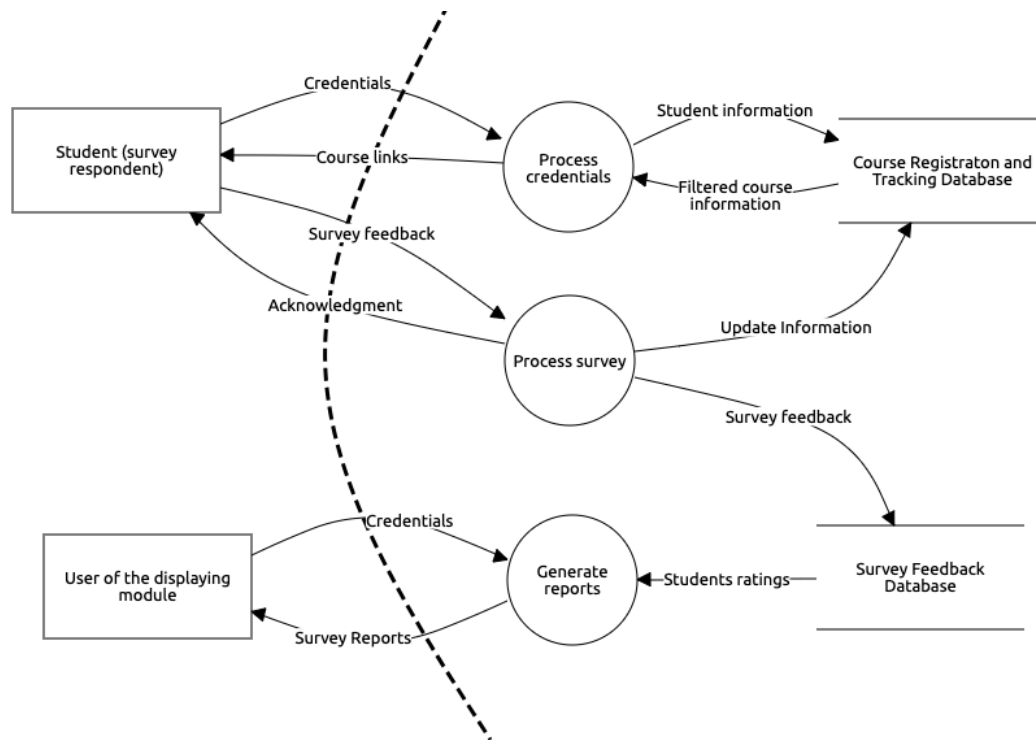
The proposed mitigation strategies for reducing the risk of linkability and identifiability in the data store are centered around data minimization: the propositions are to store only the necessary amount of data and to conduct regular audits of the data store to update and remove data which is no longer needed; avoid storing redundant data; do not store students' quasi-identifiers during the submission of survey questionnaires without the students' knowledge and consent; conduct regular security audits to ensure that the access to the database is secure.

## 6.3 Suggestions for enhanced anonymity

This section presents additional ideas, that would enhance the anonymity of survey responses but which would not be as straightforward to incorporate into the current version of the application and would require more fundamental architectural changes to the survey system. Therefore, it serves more as inspiration and a stimulus for discussion about the future development of the CTU Teaching Survey.

The anonymity enhancements presented in this section come from a research paper “An information system architecture for ensuring anonymity of student survey responses” published in *The International Journal of Information and Learning Technology* in 2019. [4] The text describes an information systems architecture of a survey system which guarantees the anonymity of student survey responses of teaching. [4]

Although the proposed anonymous survey system differs from Anketa CTU in certain aspects (such as the technologies used), the main principles for ensuring student’s anonymity presented in the paper are a useful demonstration of countermeasures against system characteristics which weaken the anonymity property and that are currently a part of the CTU Survey application architecture.



■ **Figure 6.4** DFD of an anonymous survey system from [4]

The DFD in figure 6.4 describes the architecture and workflow of the anonymous survey system. At the start of the filling phase of the survey students receive links to the questionnaires for courses they are registered for. To access the links, students are required to log in with their university ID and password. The system processes their credentials and retrieves the specific questionnaires which the student is eligible to submit from the course registration and tracking database. [4]

To guard the association between respondents’ PII and their anonymous survey responses after submission the article [4] proposes to have two separate databases. The first database contains data related to student course registration information (such as student name, student identification number, demographic information), as well as course information (course title, course number, semester, enrollment, etc.). The second database is used for anonymized student responses excluding all information about students’ PII or any association with students’ identity. This ensures that the stored student information and survey responses are impossible to link even for the survey administrator. [4]

When the student completes and submits the survey, the system simultaneously updates the student’s record in the course registration and tracking database and sends the anonymized

survey responses to the survey feedback database for further processing. [4]

During the survey publication phase, the reports from the survey become accessible to authorized personnel who need to provide valid credentials to log into the application and view the generated reports. [4]

The anonymity-ensuring features of the proposed architecture adhere to the LINDDUN taxonomy of mitigation strategies from figure 6.1. The anonymity system design ensures both the concealment of associations, by separating the filling and displaying parts of the application interface and processes to conceal the association between users and their actions, as well as the guarding of associations after submission, by saving the anonymous and identifiable information into two separate databases.

## 6.4 Recommendations for students

The following section aims to provide a sum up of the information derived from the anonymity analysis in a more user friendly way. It is mainly dedicated to the CTU students and other users of the CTU Teaching Survey who would like to know more about the application's inner workings and about the implementation of the anonymization of survey responses. It also provides recommendations for users for approaching their privacy and security when using the Anketa CTU application (and web applications in general).

### 6.4.1 How does the application ensure anonymity?

Anketa CTU is a non-anonymous application since users have to use their CTU username and password to log into the application. The only anonymous feature is the submission of student feedback. All the other features provided within the application are not anonymous. For example, the Slack feedback tool attaches the student's username to their feedback message, and the Sentry application used to report errors and crashes in Anketa CTU also attaches the current user's username to the error reports.

When a student fills and submits their questionnaire, a request is created containing their survey response as a payload and sent from their browser over the internet to the Anketa CTU server. The communication between the client and server is secured via the HTTPS protocol.

There is an authentication token (JWT) sent with each request from the user's browser to the Anketa CTU server. This token is unique for each user and is cryptographically signed to verify the user's requests and also to stop anyone from tampering with the token. If someone tried to change the token payload, the server would recognize that the signature is invalid and reject the incoming request.

After the server validates the incoming request, the survey questionnaire is anonymized by storing the content separately from the student's identity in the application database. The anonymous survey answers are stored in one database table in which the student's name is replaced with the answer id. The information about the event (that the student already submitted their feedback for the course) is saved in a separate table together with the student id and a timestamp. Access to the database is restricted to a limited group of people, including the survey and database administrator, application developers, and testers.

The analysis did uncover minor issues with the anonymization process. Although the mechanism itself separates the student identity from their responses, it is still possible in some cases for a person with access to the application database to link pieces of information from several tables and re-identify the original respondent. This is due to insufficient data minimization. There is a redundant set of information saved to the database with each student's survey answer, making it easier to reconnect the survey answer to the student's identity. This does not mean that the process of re-identification is straightforward and applicable to each survey respondent and response, but it is not completely impossible.

## 6.4.2 Privacy recommendations

The use of the public CTU usernames for authentication makes it easier to launch brute force attacks and dictionary attacks to guess the user's password. Therefore the first piece of advice for the users of Anketa CTU is to use a strong password, which is not easily deducible and to change it regularly. [37]

The next recommendation for the users is to always make sure that the website to which they are about to input their credentials or other sensitive information uses HTTPS protocol (as opposed to the insecure HTTP protocol) as well as valid TLS certificate. Typically, browsers will display a padlock icon in the address bar to indicate that a site is using HTTPS and that the connection is secure. Some browsers may also display a warning message or an exclamation mark icon if a site is using HTTP or does not use a valid certificate and may be potentially insecure (e.g. in figure 5.1).

Regarding the submission of the teaching survey questionnaires the first recommendation for students is to pay attention to the anonymity panel before submission of survey answers (figure 5.11) to make sure that they do not disclose more information about themselves than they wished to disclose.

Another recommendation for students is to be cautious when providing qualitative feedback to not include information that might make them identifiable.

The analyses uncovered that the CTU Survey uses Google Analytics service to monitor users' behavior on the site. At the moment the application does not provide an option for users to block the GA cookies from being sent to the GA server. There are however some solutions that the users can use to block the Google Analytics cookies from being sent from their browser. The tool that I tested<sup>1</sup> and found to be effective for this purpose is the "Google Analytics Opt-out Add-on (by Google)" extension. Users can add this extension to their browser to stop websites from sending information to Google Analytics. [38]

---

<sup>1</sup>The results from the test of the Google Analytics Opt-out Add-on extension on Anketa CTU are included in the electronic attachment of this thesis.

# Conclusion

The objective of this thesis was to conduct a threat analysis of the CTU Survey application's mechanisms that ensure anonymity of survey respondents. The chosen frameworks to guide the threat analysis process were the LINDDUN privacy threat modeling framework and the DREAD threat evaluation model. The application's source code and architecture were studied to gain a solid understanding of the system's inner workings.

The first three chapters of the thesis provided an overview of the application under analysis, along with the theoretical concepts of anonymity and privacy and their significance in the context of the CTU Teaching Survey. The reader was introduced to the topic of threat modeling and the LINDDUN threat modeling methodology, which was chosen as the main method for the anonymity analysis of the CTU Teaching Survey system.

The second half of the thesis consisted of three core chapters presenting the approach to the anonymity analysis, results, and recommendations derived from the analysis. The process of applying a threat modeling methodology to an existing system was described, including the creation of data flow diagrams of the application, showcasing its different components, actors, data flows, and trust boundaries. The model was used to elicit and document privacy threats, utilizing the LINDDUN knowledge base, as well as the LINDDUN catalog of privacy threat trees. The threats were ranked based on the risk they posed to the application security and the respondent's anonymity, using a combination of the DREAD ranking method and the LINDDUN threat categories. The resulting ordered list of threats was then used to search for manifestations of the listed threats in the actual system in the form of vulnerabilities, shifting focus from the application diagrams to the system itself.

The anonymity analysis uncovered several issues withing the application, which could have a negative impact not just on the respondents' anonymity but also on the overall system's confidentiality and integrity. These findings included a serious security misconfiguration of the system, non-compliance issues as well as other privacy related issues. The final section of the practical part of the thesis proposed potential mitigations for the identified system flaws and provided recommendations for the survey respondents.

I encountered some limitations during the analysis which inspired me to include propositions for further extensions of my work. LINDDUN as a privacy threat modeling methodology was designed to be incorporated in the secure development lifecycle and to be performed together with (or prior to) a security threat analysis using STRIDE. My suggestion would be to conduct a full security analysis of the application as a follow up to the privacy analysis. For the purpose of the security analysis the analyst would be able to leverage and update the system description, as well as the threat model of the CTU Survey provided in this work.

The main contribution of the thesis was the outcome of the privacy analysis of the CTU Teaching Survey application which managed to uncover issues within the system and also provided

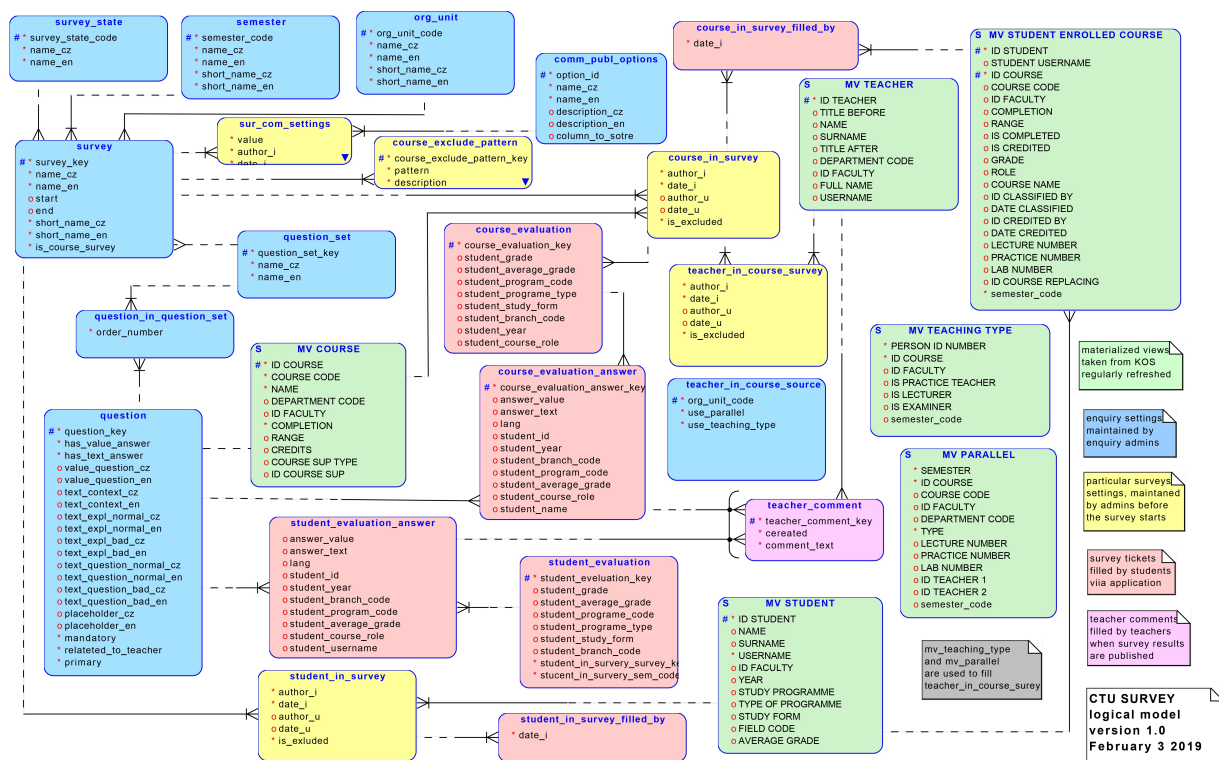
suggestions for their mitigation.

However, the objectives of this work were not limited only to searching for possible pitfalls and errors in the application which could threaten to compromise the survey respondents' anonymity (even though that was my main concern as the analyst), but also to provide the reader with a description and better understanding of the overall system and to increase the transparency of the system and promote its use among CTU students. This work also has the potential to contribute to the users' general awareness about privacy and anonymity and to the further development of the CTU Survey application.

In broader scope this work aims to contribute to the existing research and discussion about privacy in online survey systems (and in software systems in general) by providing an example of one possible way of applying the LINDDUN threat modeling methodology to an existing system.

# Appendix A

## Database ER model [6]







## Appendix B

# Risk evaluation tables

■ **Table B.1** DREAD risk rating for each threat applicable to Anketa CTU

Privacy threat	D	R	E	A	D	total	priority
Use of production data for testing purposes in publicly accessible test environments	2	3	2	3	2	12	high
JWT token misconfiguration	3	3	3	3	2	14	high
Missing consent cookie banner	1	3	1	3	3	11	medium
Weak implementation of separating student's identity from Survey responses	3	3	2	2	1	11	medium
Linkability of survey responses	2	2	2	2	2	10	medium
Linkability of survey responses for one course	2	2	2	2	2	10	medium
Identifiable login of CTU Survey users (guessed and used in the test env.)	2	3	3	3	2	10	medium
Identifiability of students as result of insufficient testing	2	1	2	2	1	8	medium
Default settings might lead respondents to disclose more data than they wished to disclose	1	3	2	2	3	11	medium
Students are not aware of the consequences of disclosing information about themselves	2	2	1	2	3	10	medium
Survey respondents are unaware of stored data	2	3	1	2	1	9	medium
Too much data stored in the database leads to linkability	1	2	2	2	1	8	medium
Storing data too long leads to linkability	2	2	1	2	1	8	medium
Re-identification of survey respondents	3	2	1	2	1	9	medium
Data identified by username	3	2	1	2	1	9	medium
Redundant data stored with each survey response	2	3	1	2	1	9	medium
Disclosure of application server version	1	3	1	1	3	9	medium
Linkability of data displayed on GUI	1	1	2	2	3	9	medium
Student's evaluation of a specific course detectable in STG env.	2	3	3	2	2	12	high

Publicly accessible STG and DEV env. lead to information disclosure	3	3	3	2	2	13	high
Survey results disclosed before intended date	1	2	3	3	2	9	medium
Student identifiability on Anketa GUI	2	1	2	2	3	10	medium
Non-repudiation of survey respondents	2	1	2	2	2	9	medium
Unencrypted communication in test env.	3	3	3	3	2	14	high
Linkability of contextual data (metadata)	1	2	1	2	2	8	medium
Weak implementation of SAML assertion	3	3	3	3	3	15	high
Weak implementation of SAR security	3	3	3	3	3	15	high
Disclosure of user credentials within LDAP request	3	3	3	3	3	15	high
Disclosure of information within LDAP verification message	3	3	3	3	3	15	high
Linkability of GA cookies content leads to inference	1	1	1	2	1	6	low
Use of GA cookies might lead to user identification	1	2	1	3	1	8	medium
Use of GA cookies leads to information disclosure	2	2	1	3	1	9	medium
Detectability of user's behaviour on the GUI	2	1	1	3	1	8	medium
Non-repudiation of actions performed by users due to GA	1	1	1	3	1	7	low
Users unaware that their survey accounts are publicly accessible	3	1	1	3	1	9	medium
Application logging enables re-identification of survey responses	3	3	2	3	2	13	high

■ **Table B.2** Ordered list of threats based on risk

Rank	Privacy threat	priority	LINDDUN category
1	Weak implementation of SAML assertion	15	Disclosure of information
2	Weak implementation of SAR security	15	Disclosure of information
3	Disclosure of user credentials within LDAP request	15	Disclosure of information
4	Disclosure of information within LDAP verification message	15	Disclosure of information
5	JWT token misconfiguration	14	Disclosure of information
6	Unencrypted communication in test env.	14	Disclosure of information
7	Publicly accessible STG and DEV env. lead to information disclosure	13	Disclosure of information
8	Application logging enables detection of survey responses	13	Detectability
9	Student's evaluation of a specific course detectable in STG env.	12	Detectability
10	Use of production data for testing purposes in publicly accessible test environments	12	Non-compliance

11	Weak implementation of separating student's identity from Survey responses	11	Identifiability
12	Default settings might lead respondents to disclose more data than they wished to disclose	11	Unawareness
13	Missing consent banner for non-technical cookies	11	Non-compliance
14	Student identifiability on Anketa GUI	10	Identifiability
15	Identifiable login of CTU Survey users (guessed and used in the test env.)	10	Identifiability
16	Linkability of survey responses	10	Linkability
17	Linkability of survey responses for one course	10	Linkability
18	Students are not aware of the consequences of disclosing information about themselves	10	Unawareness
19	Re-identification of survey respondents	9	Identifiability
20	Data identified by username	9	Identifiability
21	Linkability of data displayed on GUI	9	Linkability
22	Disclosure of application server version	9	Disclosure of information
23	Survey results disclosed before intended date	9	Disclosure of information
24	Use of GA cookies leads to information disclosure	9	Disclosure of information
25	Non-repudiation of survey respondents	9	Non-repudiation
26	Survey respondents are unaware of stored data	9	Unawareness
27	Users are unaware that their survey accounts are publicly accessible	9	Unawareness
28	Redundant data stored with each survey response	9	Non-compliance
29	Identifiability of students as result of insufficient testing	8	Identifiability
30	Use of GA cookies might lead to user identification	8	Identifiability
31	Too much data stored in the database leads to linkability	8	Linkability
32	Storing data too long leads to linkability	8	Linkability
33	Linkability of contextual data (metadata)	8	Linkability
34	Detectability of user's behaviour on the GUI	8	Detectability
35	Non-repudiation of actions performed by users due to GA	7	Non-repudiation
36	Linkability of GA cookies content leads to inference	6	Linkability



..... Appendix C

# Contents of the electronic attachments

- /
- └ thesis/.....source files of the thesis in the L<sup>A</sup>T<sub>E</sub>X format
- └ BURP-http-requests/ ..... HTTP requests captured in client-server communication
  - └ internal-login/ ..... internal login requests
  - └ LDAP-login/ ..... LDAP login requests
  - └ SSO-login/ ..... SSO login requests
  - └ questionnaire/ ..... submission of a survey questionnaire
  - └ sentry-api-error ..... error response from the Sentry server
  - └ sentry-breadcrumbs.txt ..... Sentry breadcrumbs in json format
  - └ slack-feedback ..... submission of a Slack feedback
  - └ test-scenario ..... GUI test without the GA Opt-out extension [38]
  - └ test-scenario-GA-opt-out... GUI test with the GA Opt-out extension enabled [38]
- └ survey-threat-model/ ..... files created using the OWASP Threat Dragon
  - └ anonymous-survey-model.json ..... DFD of the anonymous survey system [4]
  - └ threat-model-of-the-ctu-teaching-survey/....threat model files of Anketa CTU
    - └ AnketaCTUmodel.json .....threat model in JSON format
    - └ AnketaCTUmodel.pdf .....threat model report
- └ thesis.pdf .....full text of this thesis in PDF



# Bibliography

1. WUYTS, Kim. Privacy threats in software architectures. 2015. ISBN 9789460189500.
2. SHOSTACK, Adam. *Threat modeling: Designing for security*. John Wiley & Sons, 2014.
3. WUYTS, Kim; JOOSEN, Wouter. LINDDUN Privacy Threat Modeling: A Tutorial. *Rep. CW 685*. 2015. ISBN 9788578110796. ISSN 1098-6596. Available from arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).
4. ARDALAN, Alireza; ARDALAN, Roya K.; RAO, Shailaja; ALEXANDER, Kay B. An information system architecture for ensuring anonymity of student survey responses. *Int. J. Inf. Learn. Technol.* 2019, vol. 36, no. 1, pp. 52–65. ISSN 20564899. Available from DOI: [10.1108/IJILT-02-2018-0011](https://doi.org/10.1108/IJILT-02-2018-0011).
5. CZECH REPUBLIC. *Předpis 111/1998 Sb., Zákon o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách)* [online]. 1998. [visited on 2023-03-26]. Available from: <https://aplikace.mvcr.cz/sbirka-zakonu/>.
6. CZECH TECHNICAL UNIVERSITY. “Anketa ČVUT” polling system: system documentation and methodology. Prague, Czech Republic, 2022. No. Čj. CVUT00018824/2022. Available also from: <https://www.cvut.cz/sites/default/files/content/d1dc93cd-5894-4521-b799-c7e715d3c59e/en/20221102-methodological-guideline-no-32022.pdf>.
7. KNAP, David. *Návrh nového řešení aplikace Anketa ČVUT*. 2018. MA thesis. Czech Technical University in Prague. Computing and Information Centre.
8. ŠTECHA, Vojtěch. *Anketa ČVUT - verze 3.0 - vyplňování anketních lístků*. 2019. MA thesis. Czech Technical University in Prague. Computing and Information Centre.
9. JUN, Jakub. *Anketa ČVUT verze 3 - modul pro správu anket - uživatelské rozhraní*. 2020. Bachelor’s Thesis. Czech Technical University in Prague. Computing and Information Centre.
10. HAI, Nam Nguyen. *Anketa ČVUT - redesign vyplňovacího modulu*. 2022. Bachelor’s Thesis. Czech Technical University in Prague. Computing and Information Centre.
11. PFITZMANN, Andreas; HANSEN, Marit. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. *Tech. Univ. Dresden*. 2010, pp. 1–98. ISSN 00031224. Available also from: [http://dud.inf.tu-dresden.de/literatur/Anon\\_Terminology\\_v0.34.pdf](http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf).
12. ABI SEN, Adnan Ahmed; BASAHEL, Abdullah M. A comparative study between security and privacy. In: *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2019, pp. 1282–1286.
13. BRAITERMAN, Zoe; SHOSTACK, Adam, et al. *Threat Modeling Manifesto* [online]. 2020. [visited on 2023-04-30]. Available from: <https://www.threatmodelingmanifesto.org/>.

14. CONKLIN, Larry. *Threat Modeling Process | OWASP Foundation* [online]. 2022. [visited on 2023-02-17]. Available from: [https://owasp.org/www-community/Threat\\_Modeling\\_Process#threat-model-information](https://owasp.org/www-community/Threat_Modeling_Process#threat-model-information).
15. DRAKE, Victoria. *Threat Modeling* [online]. 2021. [visited on 2023-04-01]. Available from: [https://owasp.org/www-community/Threat%7B%5C\\_%7DModeling](https://owasp.org/www-community/Threat%7B%5C_%7DModeling).
16. MOHANAKRISHNAN, Ramya. *Top 10 Threat Modeling Tools in 2021* [online]. 2021. [visited on 2023-04-02]. Available from: <https://www.spiceworks.com/it-security/vulnerability-management/articles/top-threat-modeling-tools/>.
17. GOODWIN, Mike; GADSDEN, Jon; READING, Leo; KOKOROKO, Arnold. *OWASP Threat Dragon*. The OWASP Foundation, 2023. Available also from: <https://owasp.org/www-project-threat-dragon/>.
18. SHEVCHENKO, Nataliya; CHICK, Timothy A; O'RIORDAN, Paige; SCANLON, Thomas P; WOODY, Carol. *Threat modeling: a summary of available methods*. 2018. Tech. rep. Carnegie Mellon University Software Engineering Institute Pittsburgh United.
19. DISTRINET KU LEUVEN. *LINDDUN* [online]. 2023. [visited on 2023-04-30]. Available from: <https://linddun.org/>.
20. WUYTS, Kim; SCANDARIATO, Riccardo; JOOSEN, Wouter. *LIND(D)UN privacy threat tree catalog*. *Department of Computer Science, KU Leuven*. 2014.
21. OWASP. *Risk rating methodology* [online]. 2023. [visited on 2023-04-18]. Available from: [https://owasp.org/www-community/OWASP\\_Risk\\_Rating\\_Methodology](https://owasp.org/www-community/OWASP_Risk_Rating_Methodology).
22. CZAGAN, Dawid. *Qualitative risk analysis with the DREAD model* [online]. 2014. [visited on 2023-04-19]. Available from: <https://resources.infosecinstitute.com/topic/qualitative-risk-analysis-dread-model/>.
23. STANGANELLI, Joe. *Which Threat Risk Model Is Right for Your Organization?* [online]. 2016. [visited on 2023-04-20]. Available from: <https://www.esecurityplanet.com/networks/which-threat-risk-model-is-right-for-your-organization/>.
24. GAËL, B. *Threat modeling: which method should you choose for your company? (Stride, Dread, QTMM, LINDDUN, PASTA)* [online]. 2022. [visited on 2023-04-20]. Available from: <https://positivethinking.tech/insights/threat-modeling-which-method-should-you-choose-for-your-company-stride-dread-qtmm-linddun-pasta/>.
25. TERAIVAINEN, Taina. *DEFINITION single sign-on (SSO)* [online]. 2022. [visited on 2023-04-21]. Available from: <https://www.techtarget.com/searchsecurity/definition/single-sign-on>.
26. CANTOR, Scott. *Shibboleth Concepts* [online]. 2022. [visited on 2023-04-21]. Available from: <https://shibboleth.atlassian.net/wiki/spaces/CONCEPT/overview?homepageId=928645504>.
27. IVANTI. *Encrypting SAML assertions* [online]. 2021. [visited on 2023-04-22]. Available from: [https://help.ivanti.com/mi/help/en\\_us/ACC/50/gd/AccessGuide/Encryption%20certificates.htm#:~:text=The%20SAML%20assertions%20are%20encrypted,Response%20itself%20is%20not%20encrypted..](https://help.ivanti.com/mi/help/en_us/ACC/50/gd/AccessGuide/Encryption%20certificates.htm#:~:text=The%20SAML%20assertions%20are%20encrypted,Response%20itself%20is%20not%20encrypted..)
28. ONETRUST. *What is an HttpOnly Cookie?* [online]. 2021. [visited on 2023-04-21]. Available from: <https://www.cookiepro.com/knowledge/httponly-cookie/>.
29. JOHNSON, Glen. *LDAP Authentication Using Pure Java* [online]. 2022. [visited on 2023-04-22]. Available from: <https://www.baeldung.com/java-ldap-auth>.
30. GRASSI, Paul A; FENTON, James L, et al. Draft nist special publication 800-63b digital identity guidelines. *National Institute of Standards and Technology (NIST)*. 2016, vol. 27.



31. DATPROF. *Test data compliance*. 2023. Available also from: <https://www.datprof.com/test-data-compliance/>.
32. OKTA, INC. *JSON Web Tokens* [online]. © 2023. [visited on 2023-04-08]. Available from: <https://jwt.io/>.
33. URRACO, Mauricio. *Implementing JWT with Spring Boot and Spring Security*. 2017. Available also from: <https://medium.com/@xoor/jwt-authentication-service-44658409e12c>.
34. SENTRY. *Sentry Breadcrumbs* [online]. 2023. [visited on 2023-04-29]. Available from: <https://docs.sentry.io/product/issues/issue-details/breadcrumbs/>.
35. CIPS. *Ta-Technika* [online]. 2018. [visited on 2023-04-29]. Available from: <https://www.cips.cvut.cz/projekty/ta-technika/>.
36. EUROPEAN COMMISSION. *Reform of EU data protection rules* [online]. 2018. [visited on 2023-04-26]. Available from: [https://commission.europa.eu/law/law-topic/data-protection/reform\\_en](https://commission.europa.eu/law/law-topic/data-protection/reform_en).
37. GARFINKEL, Simson L. NISTIR 8053 De - Identification of Personal Information NISTIR 8053 De - Identification of Personal Information. *Natl. Inst. Stand. Technol.* 2015, p. 54.
38. GOOGLE. *Google Analytics Opt-out Add-on (by Google)*. ga-extension-publishers, 2021. Version 1.1. Available also from: <https://chrome.google.com/webstore/detail/google-analytics-opt-out/fl1aojicojecljbmefodhfapmkghcbnh>.