TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Cyber Security

IVCM09/14
Urmo Lihten 143912IVCM

DETECTION OF PROCESS ABUSE AND DATA REQUEST MISUSE ON ELECTRONIC HEALTH RECORD SYSTEM BASED ON REQUEST LOGS

Master Thesis

Supervisor: Firstname Lastname PhD
Co-Supervisor: Firstname Lastname MSc

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutitehnika instituut

IAY70LT
Firstname Lastname 123456 ABCD

PROTSESSI KÕRVALEKALLETE JA ANDMEPÄRINGUTE VÄÄRKASUTUSE AVASTAMINE TERVISE INFOSÜSTEEMIS PÄRINGU LOGIDE PÕHJAL

Magistritöö

Juhendaja: Firstname Lastname PhD
Kaasjuhendaja: Firstname Lastname MSc

Tallinn <year>

Author's declaration of originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. All works and major viewpoints of the other authors, data from other sources of literature and elsewhere used for writing this paper have been referenced.

Author: Urmo Lihten

April 14, 2020

Abstract

Here goes your abstract...

The thesis is in English and contains 18 pages of text, 5 chapters, 23 figures, 8 tables.

Annotatsioon

Annotatsioon on lõputöö kohustuslik osa, mis annab lugejale ülevaate töö eesmärkidest, olulisematest käsitletud probleemidest ning tähtsamatest tulemustest ja järeldustest. Annotatsioon on töö lühitutvustus, mis ei selgita ega põhjenda midagi, küll aga kajastab piisavalt töö sisu. Inglisekeelset annotatsiooni nimetatakse Abstract, venekeelset aga

Sõltuvalt töö põhikeelest, esitatakse töös järgmised annotatsioonid:

- kui töö põhikeel on eesti keel, siis esitatakse annotatsioon eesti keeles mahuga $\frac{1}{2}$ A4 lehekülge ja annotatsioon *Abstract* inglise keeles mahuga vähemalt 1 A4 lehekülg;
- kui töö põhikeel on inglise keel, siis esitatakse annotatsioon (Abstract) inglise keeles mahuga $\frac{1}{2}$ A4 lehekülge ja annotatsioon eesti keeles mahuga vähemalt 1 A4 lehekülg;

Annotatsiooni viimane lõik on kohustuslik ja omab järgmist sõnastust:

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 18 leheküljel, 5 peatükki, 23 joonist, 8 tabelit.

Glossary of Terms and Abbreviations

ATI	TTÜ Arvutitehnika instituut
DPI	*Dots per inch*, punkti tolli kohta

Contents

List of Figures

List of Tables

1. Introduction

Estonian Health Information System gives doctors the ability to send patient data to centralized information system. From there other medical workers and also the patient can view entered documents and information. This also gives doctors and nurses access to private data, when they are doing examinations and other procedures to their patients.

1.1. Problem statement

Since medical staff can view peoples medical historyin Health Information System, this poses security threat of misusing the queried data. All that is needed, to see the information, is the persons identification code.

It is hard to determine, if patient has really turned to them for medical help or not. When in an emergency and patient is un-cooperative or in such state unable to communicate, patients identity has to be confirmed without an persons consent. Permission or rights to view patients data is usully given, when the person turns to the doctor with medical issue. Meaning, the permission is not spefically given in the information system and thus allowing view data knowing just the persons personal This allows medical staff to open any persons medical history and view it at any given time whether the person has any medical relationship to that medical staff or does not.

When a persons private data such as medical information is viewed and used, then there has to be a reason. Even if it wrongly done but is still explainable (wrong identification code submission into the system by accident due to similarities, typing wrongly by mistake or third person has given wrong patient identification code by accident).

To solve this problem is to detect health records data misusage and errors in the process as early as possible by analyzing Health Information System logs what include data requests and documents sent. Learning about the different processes in which different queries has to be made within patient treatment and data forwarded. This gives the possibility to detect processes and its anomalies - queries and documents sent when not needed or out of the ordinary. Upon problem detection healthcare service provider can be contacted and be questioned, if action was intententional or not. Also to find out the reason. If the misuse is very serious, proper action has to be taken by proper authorities.

2. Approach overview

Estonian Health Information System logs every data query and document sent to it as requests. Every response is either data from queried documents and/or from subqueries to other data providers or approvement, that document or data is saved. Before the request reaches to the database, there are multiple layers of services that receive the request and examine it, if the organization is allowed to send it, if its properly constructed to its standard, if the syntax of query is valid and if subqueries to other information systems is needed. When some part of the checks and validations fail to accept the query, proper error message is sent as an response. If data query is too large or query requests data for large period then information system might cancel the query if it takes a long time to respond or its unable to respond. Requests and responses are sent as XML SOAP messages. These contain different object identification codes to classify each document and query.

Usually every request is made following a certain process. This is agreed upon on an organizational and national level. In information system the process model might be different and needs to be found out. For this process mining tools and machine learning techniques could help to create process models and check conformance. Also detect anomalies is data usage.

Every request has to be parsed for certain data fields, which give input for the process mining tools to form a process model. After that, a conformance check can be made to find anomalies and machine learning helps to find out data misusages.

2.1. Data cleansing

Logs have large amount data and not every piece of it is needed. These have to be cleansed and selected what to use with machine learning algorithms. If data is doubled (same thing but different representation) then the doubled data does not give anything new value to analyze and learn. Other data, that does not help the inital goal, also is not valueable.

Goal is to detect abuse and misuse of data requests. These requests have standardized fields of objects and their values according to X-Road request and response structure and international electronic health record HL7 standard.

To avoid any friction of the data protection law and persons private data, selection of data fields is chosen. This is conformed with Health and Welfare Informations Systems Centre information security specialists and ethics commitee in Ministry of Social Affairs.

Selected data include health care organizations national registry code, healthcare organization workers identification code (who made the request), request type, response type, request timestamp, response timestamp, document type, sent document number (anonymized), responsed document number(s) (anonyimized), document forwarded timestamp, patients identification code (anonymized).

Organizations and persons Identification and document numbers are needed to maintain relationships between differents requests and chain together requests and documents queried or sent and form a process model based on the data. This helps to find differences in process models used by organizations and give insight what could be better or data is being used.

Anonymized data is generated to hide any visiable and person identifiable information from the logs since we do not need to find out specific persons data - we have to maintain the requests and responses relationships to a process model done for specific person or their medical case what needs to be conformed and any misuse should be identifiable.

Data cleansing is done by pulling logs and parsing them through and extracting required data fields to a table format

2.2. Data representation

Gathered data is saved in a table format Python module Pandas dataframe (similar to excel spreadsheet or CSV file). Every row describes a log entry and column represents log entries attributes what have been previously extracted and were limited to in regards of the information protection law and information security requirements.

2.3. Methods for parsing logs for needed data

Logs are collected, viewable and searchable through ELK stack (ElasticSearch, Logstash, Kibana). From x-Road servers each data request and response is sent to a centralized logging system called Logstash. After processing the incoming log streams, data is stored (or stashed) in ElasticSearch for being searchable. Kibana allows to do analytics and graphs based on that previously stored data.

Data is stored in a JSON format. Unfortunately X-road requests and responsed are in SOAP XML format which means there is XML code in JSON. XML has to be extracted out from a JSON array from a specific position. After extracting, XML has to be parsed to get needed values and place these in a dataset in a table. Keeping in mind, that in the JSON part from ElasticSearch has other metadata fields for the log line and is also needed to form a process model and chains for the happend processes.

Python scripting is used to connect ElasticSeach API and pull data from an specific index related to X-road security server logs. Each log row is going to be parsed and 'message' column contains the most valueable part of the log row - the request or the response , what has been sent through. Other parts are also relevant - timestamp, is it a reponse log row or not, what service is being queried and what organization did the query.

Dataset example from ElasticSearch first query is imaged below and after eliminating unnessesary columns called "tags" and "archived" which are logging system specific values and do not provide additional value to the query logs. Other values are time (when it came in to the logging server), is it a response log row or not, query subsystem code (what service or system made that query), query identification code, timestamp of the query or response, message (payload of the query or response), memberclass (what type of organization made the query - national organisation or government entity), version of the document in the logging server, id of the query in the logging system. hostname (from which security server or cluster it came from, membercode as the organization registration number in Estonia).

	time	response	subsystemcode	queryid	@timestamp	message	memberclass	@version	id	hostname	membercode
0	1584316744441	True	digilugu	c11-test	2020-03-16T00:00:00.092Z	<?xml version="1.0" encoding="UTF-8"?>\n<SOAP-...	GOV	1	2287117	xrd-mlog-tst	70009770
1	1584316745283	False	digilugu	c11-test	2020-03-16T00:00:00.092Z	<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://sche...	GOV	1	2287118	xrd-mlog-tst	70009770
2	1584316800104	False	digilugu	tis2tam:20200316.020000066	2020-03-16T00:00:00.092Z	<?xml version="1.0" encoding="UTF-8"?>\n<soape...	GOV	1	2287119	xrd-mlog-tst	70009770
3	1584316744307	False	tam	c11-test	2020-03-16T00:00:00.114Z	<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://sche...	GOV	1	8564343	xrd-mlog-tst	70008799
4	1584316744435	True	tam	c11-test	2020-03-16T00:00:00.114Z	<?xml version="1.0" encoding="UTF-8"?>\n<SOAP-...	GOV	1	8564344	xrd-mlog-tst	70008799
5	1584316745303	False	ra-kaitlejad	c11-test	2020-03-16T00:00:00.114Z	<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://sche...	GOV	1	8564345	xrd-mlog-tst	70003477
6	1584316745646	False	ra-kaitlejad	1584316745381	2020-03-16T00:00:00.114Z	<soap:Envelope xmlns:soap="http://schemas.xmls...	GOV	1	8564346	xrd-mlog-tst	70003477
7	1584316744804	False	talis	b551911b-6e5a-11d2-f135-73cae11f18bf	2020-03-16T00:00:00.114Z	<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://sche...	GOV	1	8564347	xrd-mlog-tst	70008799
8	1584316748447	True	talis	b551911b-6e5a-11d2-f135-73cae11f18bf	2020-03-16T00:00:00.114Z	<?xml version="1.0" encoding="UTF-8"?>\n<SOAP-...	GOV	1	8564348	xrd-mlog-tst	70008799
9	1584316748497	False	talis	f3e405bf-8270-26ec-4d52-70453312c1ac	2020-03-16T00:00:00.114Z	<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://sche...	GOV	1	8564349	xrd-mlog-tst	70008799

Figure 1. first dataset example from elasticsearch to python pandas dataframe

Actual query and response part of the log row (the column named "message") is in XML code and some needed values for process modeling can be get only from there. This also provides some diffeculties since the queries can contain simple XML parameters and others contain large XML codes in an SOAP XML envelope. Meaning parsing through different types of queries of different services might require multiple XML code parsing loops before needed value can be found and extracted for process mining to have basis to work on.

Event names in the Estonian Health Information System (leaving out the external services that are being used to create some of the health record documents) are described with XML template ID codes which corresponds to the HL7 standardized documents. Descriptions to these are published by Health and Welface Information Systems Centre's publication center webservice located at http://pub.e-tervis.ee .

Following is a dataset example to which log parsing and needed value extraction has to reach, to put togeter an event log for process mining procedure.

timestamp	response	member	identification	subsystem	document/query type	query id	document id sent	document id(s) r
2020-03-16T00:00:01.092Z	False	70009770	EE11111111111	digilugu	1.3.6.1.4.1.28284.6.1.1.172	b551911b-6e5a	-	-
2020-03-16T00:00:02:092Z	True	70009770	EE11111111111	digilugu	1.3.6.1.4.1.28284.6.1.1.173	b551911b-6e5a	-	2020031600000
2020-03-16T00:00:05.092Z	False	70007446	EE11111111112	hksos	1.3.6.1.4.1.28284.6.1.1.169	17fad00b-13b0	20200315235921	
2020-03-16T00:00:07.324Z	True	70009770	EE11111111112	digilugu	1.3.6.1.4.1.28284.6.1.1.49	17fad00b-13b0	-	2020031523592

Table 1. TO BE dataset example

3. Discovering process models

3.1. Clustering request types

4. Conclusion

References