Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Bachelor Thesis Exposé

Privacy implications of exposing Git meta data

presented by

Arne Beer
born on the 21th of December 1992 in Hadamar
Matriculation number: 6489196
Department of Computer Science

submitted on December 11, 2017

Supervisor: Dipl.-Inf. Christian Burkert
Primary Referee: Prof. Dr.-Ing. Hannes Federrath
Secondary Referee: Prof. Dr. Dominik Herrmann

Contents

1 Motivation 3

2 Goals 4

3 Chapter Outline 6

4 Current State of Research 8

1 Motivation

Git is a code version control system which is used by most programmers on a daily basis these days. According to the Eclipse Community Survey about 42.9% of professional software developers used git in 2014 with an upward tendency [1]. It is deployed in many if not most commercial and private projects and generally valued by its users. It allows quick jumps between different versions of a project's code base and to manage and merge code from different sources to one upstream.

Several million users send new commits to their Git repositories every day. On Github alone, the currently biggest open source platform, there exist about 25 million active repositories, a total of 67 million repositories and about 24 million users [2].

Some well known projects and organizations use Git, for example Linux[3], Google[4], Adobe[5] and Paypal[6]. Every repository contains the complete contribution history of every contributing user. Each commit contains the full directory structure, a link to a blob for every file, a timestamp, a commit message from the author and more additional metadata.

This raises the question how much information is hidden in the metadata of a Git repository and which attack vectors could be introduced by mining this information, regarding a contributer or the owner of the repository.

The newly gained knowledge could be utilized by employers to spy on their employees. It could be used by an unknown attacker who aims to obtain sensitive information about a company and its employees trough their open-source projects. It is even possible that a privat person wants to monitor another person that regularly contributes to open-source repositories.

As there have not been any papers published about this specific topic or at least no public paper and Git plays such a crucial role in todays information technology, I want to investigate and evaluate this potential threat.

[1] Ian Skerret. Exclipse Community Survey 2014 Results. `https://ianskerrett.wordpress.com/2014/06/23/eclipse-community-survey-2014-results/` Retrieved Oct. 25, 2017

[2] The State of the Octoverse 2017, Retrieved Oct. 25 2011, `https://octoverse.github.com/`

[3] `https://github.com/torvalds/linux`, Retrieved Nov. 24 2017

[4] `https://github.com/google`, Retrieved Nov. 24 2017

[5] `https://github.com/adobe`, Retrieved Nov. 24 2017

[6] `https://github.com/paypal`, Retrieved Nov. 24 2017

2 Goals

The ambition of this thesis is to find possible attack vectors for knowledge extracted from git metadata of a single or multiple git repositories and analyze the possible damage potential. For that purpose I will look at three realistic attacker models and try to get as much compromising and harmful knowledge for the objective of each specific model.

In the following three different attacker models with potential goals are listed. Some goals will probably be extended, changed, added or removed during the research process.

Employer The employer tries to get as much information about its employees with the intention of spying on them:

- Direct comparison of productivity between employees
- Compliance of working hours
- Check if employees work on external projects during working hours
- Code quality between employees

Industrial Espionage The attacker tries to get as much information from the public open-source projects of a company:

- Company members
- History of all employees
- Overall status of a project
- A project's Code quality
- Internal team structures

Individual Somebody tries to get as much information about the personal life of a contributing individual:

- Sleeping rythm and daily routine
- Interests
- Programming languages and skills
- Personal relationships between various programmers

- Sick leave and holiday

To achieve these goals, a program will be developed, which gathers and preprocesses data and stores it inside a PostgreSQL database for easy access and querying. The data needed to validate our data mining models will be provided by scraping Github and open-sourced projects. The tool will thereby be able to gather all repositories from a specific person as well as all followed and following user repositories utilizing the Github API v2.

Hereafter the data is going to be further analyzed and processed for easy display of the results. If there is enough time it is planned to create a server-client application with an web interface for easy usage. The aggregated Data will be checked for possible attack vectors and a prototype for each of them will be build. The data mining techniques used for this analysis will be mostly based on The Kaufmann Series data mining book [1]. For every gained piece of knowledge an analysis for possible malicious usages will be performed and the severity of each usage checked.

After the analysis, the focus will shift to the legal justification for the gathering and processing of git metadata. Furthermore I will examine if the exposure of git metadata in open source projects is actually legal. The legal foundation for this investigation will be the German ‘Bundesdatenschutzgesetz’.

As the author of this thesis is not trained in legal questions, the legal section will not be as detailed and profound as could be expected from a law student.

3 Chapter Outline

In the first chapter I will state the general problem as well as the thesis motivation. An overview of the current project status and the goals of the thesis will follow together with the chapter outline.

Chapter two will attend to the aggregation of data. In this context I will elucidate git and go into the fundamental technology behind it, as well as its mechanisms for saving, versioning and organizing Data. The tree like structure of the git history will be investigated and some problems of this structure regarding continuous data mining of git commit histories examined. If, for example, a contributer uses force-pushes changes and thereby rewrites the git history tree, the data needs to be update accordingly and the old history has to be identified and truncated. As we get the data for the evaluation of this research from Github, the used techniques for getting data through the Github APIv2 will also be introduced.

The third chapter will discuss the general structure of the mined data and the technologies used for data preprocessing. Possible risks of unclean or imprecise data and general problems with the data will be a topic as well.

In the fourth chapter the gained knowledge and the used data mining techniques will finally be presented. The possible damage potential of this knowledge will be evaluated as well. For this purpose the correctness of the gained knowledge will be examined using the data from Github. These correctness checks probably won't be perfect, as it is hard to get real world information about employees from a company. As an example, organizations don't necessarily expose all their employees on Github and you can't look at Github organization teams without explicit access. However, knowing only a subset of employees should be enough to see tendencies and verify the correctness of an employee network analysis.

Evaluation of easier statistics like working hours and code quality can be easily verified, as they are mainly visualization of simple data. The comparison of the quality of work between two contributers can be manually checked for correctness on any repository by looking at the respective contributions. Holiday and sick leave analytics can be verified by comparing known reoccurring patterns in which people take one's holiday like Christmas, New Year's Eve or Easter.

The biggest expected problem on those easier statistics is detecting outliers such as massive changes in line of code additions caused by the inclusion of external library code.

The fifth chapter will be about the legal justification of this project and clarify if it is allowed to use this tool in a commercial environment. It will also raise the subject for the need of obscuration of git meta data in commercial open source projects.

Chapter six will contain the conclusion of the research as well as an ethical discussion about git meta data mining.

4 Current State of Research

At the moment the data aggregator prototype is functional and it is already possible to get all repositories and their data from a specific Github user, a Github user including their Stars/Followed/Followers or to just scan a arbitrary clonable git repository. There are some edge cases as the ‘force push’ and a resulting restructuring of the git history which needs separate handling. Despite that the aggregator is fully functional. The ORM models for the data and the storage in a PostgreSQL database are working and tested as well.

Bibliography

[1] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques, 3ED*. Morgan Kaufmann, 2013. ISBN: 978-9380931913.