MAKERERE UNIVERSITY
FACULTY OF COMPUTING AND INFORMATICS TECHNOLOGY
SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
BACHELOR OF SCIENCE IN COMPUTER SCIENCE
YEAR 2
BIT 2207 RESEARCH METHODOLOGY
Course Work:LITERATURE REVIEW

NAME	REG NO	STD NO
KAZOOBA B LAWRENCE	16/U/5830/PS	216004505

SAFE BROWSING

Prepared by:KAZOOBA B LAWRENCE

Lecturer:MR ERNEST MWEBAZE

9th MARCH 2018

1 Abstract

Safe Browsing is a Google service that lets client applications check URLs against Google's constantly updated lists of unsafe web resources. Examples of unsafe web resources are social engineering sites (phishing and deceptive sites) and sites that host malware or unwanted software.

2 Introduction

In 2008, Google started a service called Google Safe Browsing (GSB) [1] to warn and dissuade an end user from visiting phishing and malware web pages. With similar goals, Yandex followed up with an identical service named Yandex Safe Browsing (YSB). As of today, all major web browsers including Firefox, Internet Explorer, Safari, Opera and Yandex. Browser feature one of these Safe Browsing (SB) services. The integration of SB services into the browsers has naturally generated an extremely large user base. GSB alone accounts to a billion users until date [2].

3 OVERVIEW

Boosted by recent legislations,[3] data anonymization is fast becoming a norm. However, as of yet no generic solution has been fouSnd to safely release data. As a consequence, data custodians often resort to ad-hoc means to anonymize datasets. Both past and current practices indicate that hashing is often believed to be an effective way to anonymize data. Unfortunately, in practice it is only rarely effective. This paper is a tutorial to explain the limits of cryptographic hash functions as an anonymization technique. Anonymity set is the best privacy model that can be achieved by hash functions. However, this model has several shortcomings.We can identify three case studies to illustrate how hashing only yields a weakly anonymized data. The case studies include MAC and email address anonymization as well as the analysis of Google safe browsing.

4 CONCLUSION

Complex URL checks You need to know how to canonicalize URLs, create suffix/prefix expressions, and compute SHA256 hashes (for comparison with the local copies of the Safe Browsing lists as well as the Safe Browsing lists stored on the server). If you are concerned about the privacy of the queried URLs or the latency induced by a network request, use the Update API.

5 OTHER OPTIONS TO CONSIDER

Check pages against our Safe Browsing lists based on platform and threat types. Warn users before they click links in your site that may lead to infected pages. Prevent users from posting links to known infected pages from your site. The Safe Browsing APIs (v3) are now deprecated and will be turned down on October 1st, 2018. All Safe Browsing API clients should use the (v4) APIs going forward.

6 REFERENCES

1."Safe Browsing API", [online] Available: https://developers.google.com/safe-browsing/.

2."Google Transparency Report", Google Tech. Rep., June 2014, [online] Available: https://bit.ly/1A72tdQ.

3.The analysis of Google safe browsing. Published in: IEEE Communications Surveys and Tutorials (Volume: 20, Issue: 1, Firstquarter 2018) Page(s): 551 - 565 Date of Publication: 31 August 2017

Electronic ISSN: 1553-877X

DOI: 10.1109/COMST.2017.2747598 Publisher: IEEE Sponsored by: IEEE Communications Society