ON PRIVACY AND THE WEB

Wesley A Brandi **Martin S Olivier**

Information and Computer Security Architectures Research Group
Department of Computer Science, University of Pretoria, Pretoria

ABSTRACT

Chor et al [3] show that when accessing a single public database, a user is only guaranteed safety from an administrator inferring the user's real intentions (an inference attack) when the user downloads the entire database. Although this approach is somewhat impractical, it is the only way in which to guarantee complete safety from prying eyes.

Inference attacks on a user generally assume that the attack is taking place from the perspective of the Database Administrator. It is therefore implicit that there is an intimate knowledge of the database being accessed by the user.

Given the nature of the Web and some of the Large Public Databases being accessed via the Web, we wish to determine if inference attacks launched on a user can be successful without detailed knowledge of the database.

Can we successfully violate the privacy of a user by analysing his Web queries to the Large Public Database over a period of time? If this is possible, how can one circumvent such an attack?

A search engine on the Web is a prime example of a Large Public Database. It is publicly accessible and users must abide by its usage policies. In this paper we discuss the issues involved in preparing to visualise a log of queries submitted to a Large Public Database. In particular, we discuss the environment in which the logs will be collected, analyse the states a user undergoes when submitting queries to a search engine and set the stage for future research.[1]

KEY WORDS

database, inference attack, privacy, search engine

[1]This material is based upon work supported by the National Research Foundation under Grant number 2054024 as well as by Telkom and IST through THRIP. Any opinion, findings and conclusions or recommendations expressed in this material are those of the author(s) and therefore the NRF, Telkom and IST do not accept any liability thereto.

1 INTRODUCTION

The primary goal of the research discussed in this paper is to determine whether the privacy of a user can be violated through careful analysis of queries made to a Large Public Database (LPD) on the Web. An LPD, in the scope of this research, offers users access to a single database on its own terms i.e. no custom (foreign) protocols can be implemented in accessing the LPD. Solutions to privacy problems that are based on custom protocols or closed systems therefore do not apply in the scope of an LPD.

Google.com is an excellent example of an LPD. Access and usage thereafter is granted on its terms only: take it or leave it. Users wishing to safeguard their privacy by encrypting the queries sent to Google simply do not have this option since Google does not cater for it.

In our research, no knowledge of the database being queried is claimed, other than what we have learnt through usage and an analysis of the way in which it is queried, as discussed in section 3.3. In analysing the queries, it is hoped that sufficient data can be gathered so as to successfully launch an inference attack on the user.

We view the concept of privacy as defined by Lategan et al [4]: *'a state that exists when access to private information about a particular individual can be effectively controlled and managed by that individual even after a third party has collected such private information.'*

In the scope of this research, the individual in question is not explicitly divulging private information. But, in analysing queries made by the individual to an LPD we wish to determine if private information can be inferred. Successfully violating the privacy of an individual is therefore defined as having obtained private information about the individual solely through analysis of his queries made to an LPD.

This paper deals with several issues preceding the analysis and visualisation of the query data that deserve attention. We believe that visualisation of the query data will provide insight into understanding what is needed in order to launch an inference attack on a user of an LPD. Visualisation will aid us in determining whether the attack can be completely automated or whether a human element will be necessary.

The need to visualise a user's query data has given rise to a simple state model specific to the LPD being queried and the given nature of the Web.

In section 2 we briefly discuss the background of privacy on the Web within the context of this research. Section 3 moves on to provide a detailed analysis of the environment under which the logs will be collected, section 4 proposes and discusses the state model in detail. This paper is concluded in section 5.

2 BACKGROUND

We have pointed out that usage of an LPD is only possible when abiding by its usage policies. No custom protocols may be implemented therefore closed solutions that employ foreign protocols to safeguard privacy (encryption schemes for example) may not be possible.

Technologies the likes of P3P [5] however, do not require the use of custom protocols. No personal information will be divulged if it conflicts with the configured privacy policy of the user. Within the scope of this research, this is already assumed. We wish to show that although

privacy information is not provided and usage of an LPD may be well within the terms of the user's privacy policy, it may be possible to violate the privacy of a user through an inference attack based on his queries and usage of the LPD.

In restricting the usage of closed system solutions when accessing an LPD, one is immediately drawn to the notion of anonymity when safeguarding one's privacy [7]. Usage of an anonymising proxy (which essentially submits an anonymous query to the LPD on behalf of the individual) to protect one's identity when accessing an LPD would hide the source of the queries from the LPD and still be in accordance to the usage policy of the LPD.

The LPD in question in this paper (Google) allows for anonymous access (as do most LPDs of this nature) therefore one would not hesitate to employ the use of an anonymising proxy when accessing the LPD. The LPD would not know who is submitting the query from behind the proxy (there can be millions of people using a proxy) rendering any type of inference attack useless. Aljifri et al [1] discuss several of the features provided by anonymising proxies to safeguard ones privacy, these include URL encoding and SSL channels to the anonymising proxy. They point out that wise usage of tools the likes of an anonymising proxy may be in the best interest of a user's privacy.

It can be argued that little trust need be associated with the proxy, since all that it is being used for is accessing the LPD. The proxy has no knowledge of the database, the database no knowledge of the individual behind the anonymising proxy. But consider the impact on a user's privacy if the proxy itself could infer private information about the user solely through analysis of queries made to the LPD.

Privacy literature acknowledges that some degree of trust must be associated with the anonymising proxy, but how much trust would one be willing to associate with an anonymising proxy that can infer private information without an individual explicitly providing it?

3 ENVIRONMENT

In this section we discuss the environment in which query data is collected. We discuss and define terms the likes of a user, a session and a query. We also examine how we track a user's activity when using Google.

The logs that contain the query data are typical to that of an Internet gateway or proxy. Figure 1 illustrates an environment where users accessing the Web (or an LPD on the Web) do so via a gateway or anonymising proxy.

In our research, the gateway used by users to access the Web is a Squid proxy server [8]. Users wishing to make use of the Web at all, need to configure their browsers to use the Squid proxy.

The following information can be gathered from the Squid proxy log when a user submits a request to access a location on the Web (submits a query to the LPD):

- The date and time the request was submitted.
- The ip address the request was submitted from.
- The Web site accessed prior to this request.

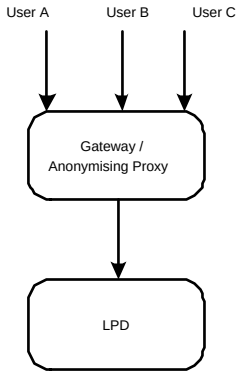

Figure 1: Typical environment used for Internet/Web Access.

- The query being submitted to the LPD.

3.1 Users

We assume that each individual user is unique based upon their assigned static IP address. Therefore, the user is essentially the IP address from which the request to access the LPD is generated. It may be argued that many users will not be accessing a proxy from the same static ip address, this may be the case particularly when one considers users that, when dialling up to the Internet are assigned a dynamic IP address, and then make use of an anonymising proxy.

In this case though, access to the anonymising proxy will probably be granted based upon some kind of authentication mechanism, a username and password, which may then be used to identify the user (as opposed to the IP address).

3.2 Sessions

Silverstein et al [6] define the notion of a session to be '*a series of queries by a single user made within a small range of time*'. We adopt the same definition of a session and use the same range of time for a session: 5 minutes. A query submitted by a user to the LPD within 5 minutes of the first query will therefore be viewed upon as belonging to the same session. Any query 5 minutes after the first query will begin another session.

3.3 Queries

A query to the LPD is first routed through the Squid proxy (where it is logged) and then forwarded by the proxy to the LPD. Queries submitted to Google consist of one or more keywords related to what the user is searching for. Google is a full-text search engine that uses Boolean logic which by default will combine all the keywords using the Boolean AND operator [2].

A simple query of 'Database Privacy' submitted to Google will have Google searching for 'Database' AND 'Privacy'. Similarly, a query of '(Database OR searchengine) Privacy' will result in Google searching for the term 'Database' or the term 'searchengine' both combined with the term 'Privacy'.

A query submitted to Google may result in Google finding one or more related sites. This result will be returned to the user via the proxy and displayed in the user's Web browser. Each link to a relevant site in the result returned by Google may contain yet another link to Google in addition to linking to the actual site concerned. This allows Google to track users in addition to allowing us to track users via the proxy logs.

3.4 Tracking

Google seems somewhat temperamental in so far as the way in which the links in a result returned from a search are served. Depending on the browser used to access Google, a link within a result may be a redirecting link (via Google) or the actual link itself used in conjunction with the loading of hidden images.

If a user conducts a search using Google with Internet Explorer 6 as his Web browser (with JavaScript enabled), it is likely that the result returned will include direct links to the sites in question. Clicking on the link however, will result in a JavaScript function being called which will load a hidden image (the embedded query) via Google.

The URL submitted to Google when loading the hidden image contains the link that is being clicked in the result, its index in the result returned in addition to other data. It is not within the scope of this research to further analyse the way in which Google tracks links that are clicked in a result returned to a user.

What is important to us, is how we determine whether a link followed by a user (logged in our proxy log) is part of a result returned by Google i.e. how do we track a user via our proxy logs? For our purposes, we assume that a user will only use one instance of the Internet Explorer 6 browser when searching via Google.

A request in the proxy log to load the hidden image via Google that follows a search query submitted by the user using Google is assumed to be a click on the relevant link of a result returned to the user.

Knowing this behaviour and given the nature of the Web (the logging facilities provided by the proxy) we have derived a state model that can be applied to each session a user is part of when using Google, as discussed in the next section.

4 STATE MODEL

What is integral to this research is that this process is conducted outside of the LPD. What we know of the LPD in question is what we have learned through usage and an analysis of the way in which it is queried. What we know of Google can be summarised as follows:

- It is a search engine whose queries are based on boolean searches.
- Each query to Google may return a set of results (this may be empty) to the user.
- A result is essentially a structure with a title, a short description and a link.
- The link provided in a result contains an embedded query to Google which will allow us to determine what links followed are part of the result returned by Google.

In logging these queries from the perspective of a proxy and in knowing the way in which this LPD queried, figure 2 depicts the state model derived.

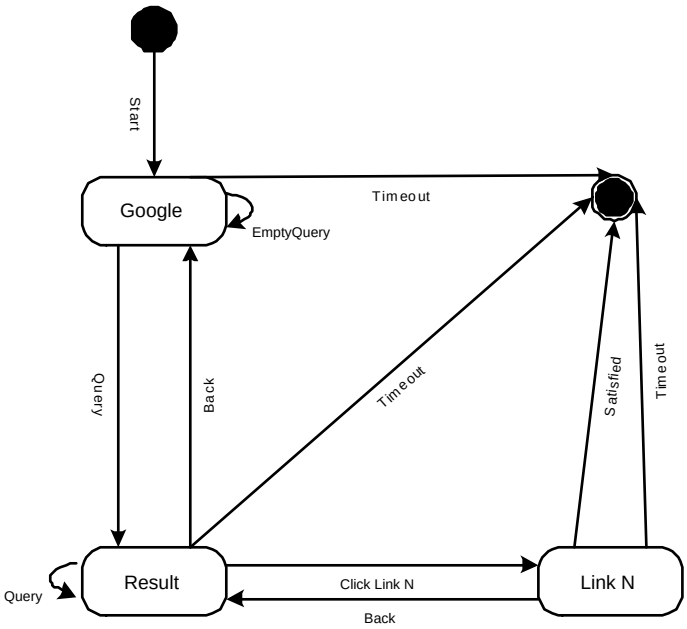

Figure 2: A state model of a session using Google.

A session begins when Google's start page (http://google.com) is accessed via a user's Web browser. The user can enter keywords into a text box on the Web page that will form part of his query. Upon clicking the Submit button, the query will be forwarded to Google. If the user entered an empty query he will be directed back to Google's start page.

A valid query will have Google return a result consisting of zero or more links. At this stage, the user can click on a link in the result or, if he is unhappy with the result, push the Back button on his Web browser to start at the beginning. Alternatively, the user can re-enter keywords into the query text box provided and search from the results page. Although searching from the results page seems as though the user is searching through the result returned, this is not the case. A search from this state will yield the same results as a search from Google's start page.

Having followed a link from the result returned by Google. We assume a user is satisfied with the link if no further links are accessed from the result for the remainder of the session or if the session timeout limit is reached. Note that there is a timeout limit associated with each state.

We do not follow the activity of a user if he accesses links not within a result returned during a session. For example, if http://link1 was part of a result returned from the LPD and the user accesses it, that is all that concerns us. Activity that includes a user following http://link1/files/ is not part of the state model since it was not a link returned from within the result.

Although this state model has been derived with Google's mechanism for searching in mind, this does not mean that the state model is specific to Google. Search engines that employ similar mechanisms for searching may also fit into this simple model. Since the logs we are using are

generated from a proxy's point of view, all Web requests made by a user are logged. All data passed from the Internet to a user via the proxy can also be logged.

This means that usage of another search engine that does not embed links within links (as Google does with the results it serves using the hidden image) can still be moulded into our state model through careful analysis of each query submitted and all data passed back to the user via the proxy.

5 CONCLUSION

The nature of the Web is such that one may need to pay more attention to services the likes of anonymising proxies when safeguarding one's privacy. What will the implications be for such services if it is possible to infer private information about a user solely through analysis of his queries to an LPD without intimate knowledge of the LPD?

A search engine has been identified as a typical LPD. We have discussed the environment in which logs of users employing the use of a search engine are being generated and stored. We define the notion of a user, a session and a query. We discuss how we will track a user's usage of the LPD solely through queries submitted and stored via the proxy. In addition to this, we present a state model to which users' sessions, when using the LPD, will be modelled. These are all issues which must be examined before moving onto the next phase of this research.

This phase will typically deal with visualising the data collected over a period of time. We believe this is an essential step in helping us to determine how an inference attack on the user is to be launched. We hope that visualisation conducted in future research will aid us in determining what the value of the actual query keywords will be within the context of this research, as opposed to more implicit results the likes of trends that are discovered over long periods of time.

References

[1] Hassan AlJifri and Diego Sanchez Navarro. Search engines and privacy. *Computers and Security*, 2004.

[2] Tara Calishain and Rael Dornfest. *Google Hacks*. O'Reilly, 2003.

[3] Benny Chor and Niv Gilboa. Computationally private information retrieval. pages 304–313, 1997.

[4] Frans A Lategan and Martin S Olivier. PrivGuard: A model to protect private information based on its usage. *South African Computer Journal*, 29:58–68, 2002.

[5] Joseph Reagle and Lorrie Faith Cranor. The platform for privacy preferences. *Commun. ACM*, 42(2):48–55, 1999.

[6] Craig Silverstein, Monika Henzinger, Hannes Marais, and Michael Moricz. Analysis of a very large altavista query log. Technical Report 1998-014, Digital SRC, 1998. http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/abstracts/src-tn-1998-014.html.

[7] Bhavani Thuraisingham, Sushil Jajodia, Pierangela Samarati, John Dobson, and Martin S Olivier. Security and privacy issues for the world wide web: Panel discussion. In Sushil Jajodia, editor, *Database Security XII: Status and Prospects*, pages 269–284. Kluwer, 1999.

[8] Duane Wessels. Squid internet object cache. http://squid.nlanr.net/Squid/.