Tracking Malware using Internet Activity Data

Abner Mendoza and Cihan Varol, *Member, IEEE*

***Abstract*— Forensic Investigation into security incidents often includes the examination of huge lists of internet activity gathered from a suspect computer. In today's age of increased internet usage, the internet activity log on any given system could produce a huge list of websites. This, couples with the fact that a huge percentage of malware is now distributed via the internet, often through compromised websites, means that valuable clues regarding the source and identity of malware infections are often hidden within the internet activity logs on a computer. While a multitude of tools exist to extract internet activity data from a host computer, most do not filter this activity data. As a result, an investigator could be faced with thousands of website URL's to sift through for clues regarding malware infection. In this paper, we discuss some of the ways that computers are infected, and why internet activity data is an important resource that must be analyzed in a forensic investigation. We then present a tool that utilizes the Google Safe Browsing Lookup API, which is an extension of the broader Google Safe Browsing API, to do quick lookups on long lists of URL's and significantly narrow the list to enable the investigator to conduct a more efficient investigation.**

Keywords- Malware Investigation, Internet Activity Data, Google Safe Browsing API

I. IINTRODUCTION

The forensic investigation process often includes the tedious and time consuming task of examining and analyzing thousands of logs and other information extracted from the system or systems in question. One such task is the process of examining web browser activity logs, either to reconstruct the user's browsing activities or to gather other clues relating to the incident under investigation. This paper is specifically focused on incidents involving malware infection. Of the many distribution vectors for malware, the most popular of late has been the use of web-based distribution. The recent Symantec Internet Security Threat Report released in 2011, shows that web based attacks increased by 36% with over 4,500 new attacks each day [1]. Other statistics gathered in the report indicate that 39% of attacks via email used a link to a webpage. Additionally, the report highlights the trend of malware distribution through otherwise legitimate websites that have been exploited to distribute malware payloads. Drive-by-downloads, as coined by Google in 2007 [2], is a particularly insidious form of malware distribution that uses browser exploits to automatically install malware on end-user machines, often without the knowledge or content of the user.

Manuscript received July 23, 2012; revised August 9, 2012.

A. Mendoza is a graduate student in the Department of Computer Science, Sam Houston State University, Huntsville, TX 77341 USA (e-mail: aam043@shsu.edu).

C. Varol is an Assistant Professor in the Department of Computer Science, Sam Houston State University, Huntsville, TX 77341 USA (phone: (936) 294-3930, e-mail: cvarol@shsu.edu).

In these cases, we rely on a database of known malware distribution websites to determine if the suspect computer has been exposed. In other cases, malware already on the computer may communicate to a home base website discreetly either for sending captured data or for downloading updates or the actual malware payload. Most of these will use built-in API's on the host operating system and artifacts of the web communication activities are saved on the host system, such as in log files and browser history files [3]. When an investigator responds to a malware incident, his forensic toolkit will usually contain one or more tools that aid in web browser forensics. In the process of examining the user's web browser activity data, an investigator could easily overlook a URL listing that may otherwise appear as a legitimate and safe website. Most forensic tools available to the investigator will stop short of analyzing the internet activity data that has been gathered from the suspect computer. It is then up the investigator to gather clues. While there are DNS blacklists and other databases available on the internet for comparing websites, most do not offer an automated tool to do a quick check of a long list of URL's. Rather, the investigator must often select individual URL's for further analysis. It is in the process of selecting these individual URL's where some portion of the listing can be overlooked. This paper proposes a tool that utilizes the Google Safe Browsing Lookup API [4] to compare a long list of internet activity data against Google's extensive database of malware-serving websites.

II. GATHERING INTERNET BROWSING ACTIVITY

The first step in the forensic analysis of web browsing history is to gather the data. There are no defined standards that dictate how browsing history is stored, and each of the major browsers utilizes different methods of storing browsing history data. For this reason, the task of gathering internet activity data is often very tedious. An investigator must be diligent and ensure that all possible data is gathered from all sources on the suspect computer. Table 1 illustrates the varying methods used by the major web browsers to store web activity history [5].

TABLE I. BROWSER HISTORY STORAGE

Browser	***Storage Medium***
Internet Explorer	Index.dat binary file
Firefox (v3.5+)	SQLite database
Chrome	SQLite database
Safari	.plist binary file *

*previously stored as XML

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Most of the tools and research that focus on extracting internet activity history have mostly concentrated on just a single browser vendor [6], but there are some efforts that have been made to devise a solution that integrates browsing history from all browsers into one analysis tool that seeks to recreate browsing time-lines and recover contents stored in cache. While this is a bit out of the scope of this paper, it is at least relevant to identify the tools and methodologies used to acquire a complete history of browsing activity regardless of the browsers used on a machine. One such tool that focuses strictly on Internet Explorer browsing history is the Pasco tool described in [3]. The history data in IE is increasingly more difficult to acquire mostly because it is stored in a binary file that is not as easily parsed as a SQLite database. Junghoon Oh, et. al. [7], specifically looks at and compares the different tools used by forensic investigators to analyze browser data. In their analysis they make note of the fact that some tools are not able to recover browsing data that have been deleted using the browser's tools to clear cache and history files. The authours introduces the WEFA (Web Browser Forensic Analyzer) tool that seeks to provide improvements to shortcomings of existing ones [7]. For the purposes of this paper, a tool such as WEFA would be the most effective tool that could provide an integrated data feed of browsing history gathered from the different browser history storage mechanisms, including information stored in cache and retrieved from deleted memory. Additionally, browsing activity data is not only available through browser history files, but also in Windows registry files, browser cookie files, cache files, and in unallocated disk space from activity files that have been deleted. To account for the varying locations from which data must be gathered, an investigator often needs to use multiple tools and then aggregate the data before examination and analysis. This inevitably leads to an increasingly larger dataset that must be examined and analyzed. Some of the popular tools for web activity forensics include FTK (Forensic Toolkit), Encase, ProDiscover, SQLitebrowser, plist Editor Pro, Pasco, and Galleta.

III. Examining Browsing Activity Data

When a browser opens a web page, the browser makes multiple requests to the host server and possibly to multiple servers across the internet to download the files and code necessary to display the web page. For example, there may be JavaScript files, images, css files, and other artifacts necessary for properly displaying web pages. Links and references to these extra files are contained in the html code of the main web page being loaded. In the process of loading a web page, the browser must fetch all these resources using an HTTP GET command. Each execution of the GET command will result in a log of internet activity. As a result, loading just a single web page could log multiple records in the browser history files pointing to various website domains, depending on the source of each of the resources that must be fetched by the browser. As an example, loading the http://cnn.com website results in 150 GET requests to fetch all the resources necessary to display the home page. Any number of these requests could potentially lead to a malicious or compromised web server, since many of them are from third party organizations such as those serving advertisements as illustrated Figure 1. Malware distributors have exploited this fact to distribute their payloads through otherwise trusted websites. For this reason, the investigator must be meticulous about examining all the activity records on the suspect computer including those that may seem harmless at first glance.

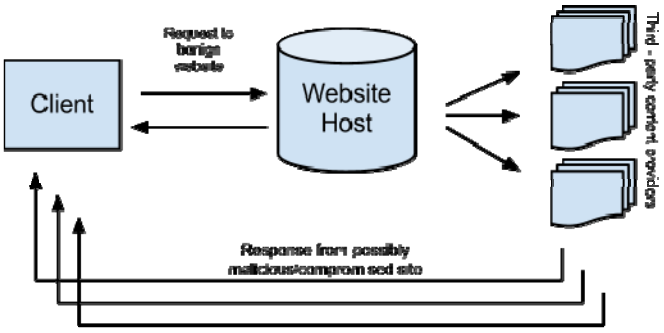

Figure 1. Typical Web Request

After gathering the browsing data from a suspect computer, it will also be necessary to filter the results and remove repeated and useless URL strings such as those pointing to static images. Given the enormity of the dataset that the investigator is likely to encounter, it is most likely not feasible to painstakingly scroll through each record manually. This is where an automated tool must be used to do the bulk of the work in eliminating those records whose probability of being malware sources are low. We can then take our filtered list of valid URL's and submit them to the Google Safe Browsing Lookup API engine for analysis. The response will indicate whether any of the submitted URLs' are known to be infected ones used to spread malware via the web. This is an immensely valuable tool to the investigator and can aid in narrowing down what could have been potentially thousands of URL's to sift through, to only a few that are tagged as malicious. The Google Safe Browsing Lookup API [4] leverages a list of constantly updated URL's suspected in phishing and malware attacks. Google's web-malware detection infrastructure, which enables this API, uses a vast array of anti-malware technologies including detection signatures, browser emulation engines, domain reputation filtering, and virtual machine client honeypots. [4] Google has been collecting data on web malware for over 5 years, and this has resulted in the release and continued improvements to the Google Safe Browsing API. The same infrastructure is enabled in the Google Chrome web browser to proactively warn a user if a site has been flagged as being malicious. Mozilla, Opera, and Apple have also forged partnerships with Google that now enabled them to also utilize the Safe Browsing API in their respective browsers. These browsers all store a form of the database locally for quick lookups before actual verification on the cloud when a URL is flagged as suspicious by the local filters [8]. Notably, Microsoft's Internet Explorer browser uses its own website blacklist to implement similar features.

IV. Safe Browsing Lookup Tool

We have developed a rudimentary tool for the purpose of testing the theory presented in this paper. Following the instructions from the Google Developer's Guide, we have developed a tool that will take a simple list of URL entries and

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

submit them in a POST request to the Google Safe Browsing Lookup API as shown in Figure 2.

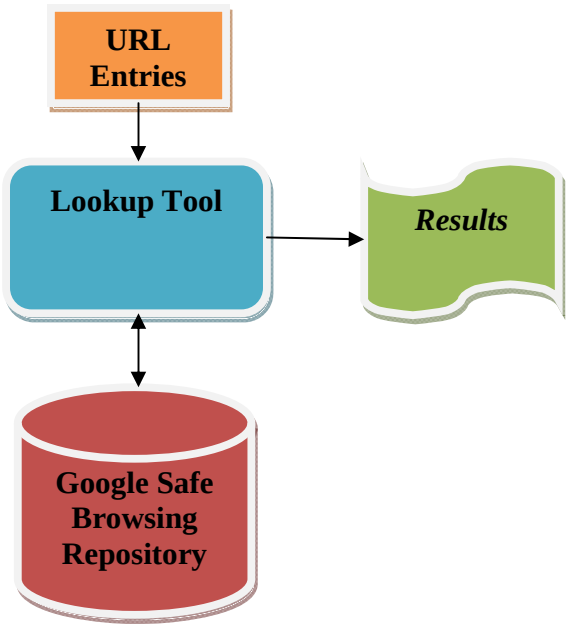

Figure 2. Safe Browsing Lookup Tool High Level Architectural Diagram

The tool will monitor the response status for an HTTP 200 response code which indicates a match in one or more of the submitted URL's. The API returns the lookup results in a simple text string that indicates "ok" for benign websites, and "malware" or "phishing" or harmful websites. Each result is separated by a line feed character encoding, it parses this into a one-dimensional array together with the original request list and displays the resulting list of URL's that have been flagged as harmful. In most cases, it is expected that the resulting list will be significantly smaller than the original list of submitted URL's. A screenshot of the tool is shown Figure 3.

In the test case shown with the screenshots, we submitted 250 sample URL listings and intentionally inserted some listings that we know to be flagged by Google. The lookup resulted in a total of 74 URLs' being flagged as malicious, which represents about 30% of our sample set. The total round trip time to send the request and obtain the result was an average of 2 seconds. This indicates that the lookup operation is quick, which is an important factor in any tool used by an investigator faced with time constraints. While there are sure to be false positives, as acknowledged by Google, it saves the investigator a tremendous amount of time in sifting through lists of URL's visited on a suspect computer. While the test case was not from a live system, we expect the similar results from history data gathered from a live system.

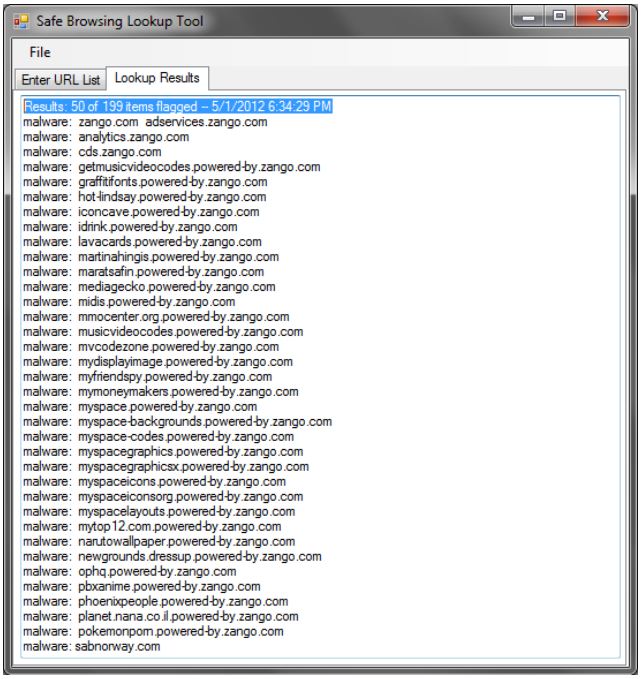

Figure 3. Safe Browsing Lookup Tool Showing Lookup Results

V. Conclusion and Future Work

Forensic Investigations will often include the extraction, examination, and analysis of internet browsing activity from a suspect computer, often as part of a broader incident investigation. When malware incidents are being investigated, the results of the browser activity logs could provide valuable clues to help identify both the source and identity of malware infections. However, the process of examining and analyzing what is often a huge list of URLs' is a tedious and time consuming task which could result in valuable clues being overlooked. The prevalence of web-based malware distribution has exploded in recent years, with malicious enterprises continuously devising new ways of exploiting vulnerabilities by using otherwise trusted websites to distribute their payloads. Even the most seasoned forensic investigator is sure to overlook otherwise benign URL's without the help of automated tools to aid in sifting through and filtering large lists of internet activity history data. We have developed a tool that utilizes the Google Safe Browsing Lookup API to quickly filter suspect URLs' from a large list, while simultaneously also taking advantage of the Google infrastructure that continuously scans the internet for malicious websites or those that have been exploited to distributed malicious payloads. Our test have shown that such an automated tool could be invaluable to a forensic investigator to both save time and identify important clues in the course of an investigation.

In this study we chose the Google Safe Browsing API simply because it is not the most widely used lookup database by virtue of its use by all major browsers except Microsoft Internet Explorer. However, there are many other lists or databases available on the web that could also be utilized. An extension of this project could include in aggregation of

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

various domain black-lists to further eliminate the possibility of a malicious URL being flagged as benign. Additionally, the tool merely lists the URLs' that are flagged, but there is no further indication of the reasons or the malware infection(s) which may have been caused by visiting the flagged URL. Google does provide a report for each URL which is accessible from http://www.google.com/safebrowsing/diagnostic?site={URL}, where {URL} represents the flagged URL. Inclusion of a link for each result in the tool would further enhance the usefulness of the tool. One drawback to using the Lookup API rather than the full Safe Browsing API is the limitation enforced by Google which only allows up to 10000 requests per day per API key. While this may suffice for simple investigations, it could also be an issue for larger deployments.

REFERENCES

[1] Symantec Internet Security Threat Report 2011, (Online) Available: http://www.symantec.com/threatreport/topic.jsp?id=highlights

[2] Provos, N., McNamee, D., Mavrommatis, P., Wand, K., and Modadugu, N. The Ghost in the Browser: Analysis of Web-based Malware. In Proceedings of the first USENIX workshop on hot topics in Botnets (HotBots'07). (April 007).

[3] Keith J. Jones, Forensic analysis of internet explorer activity files, Foundstone 2003. (Online) Available: http://www.mcafee.com/us/resources/white-papers/foundstone/wp-pasco.pdf.

[4] Google Safe Browing Lookup API, Google Inc, (Online) Available: See http://code.google.com/apis/safebrowsing

[5] Bursztein, Elie, et. al., "Doing Forensics in the cloud age. OWADE: beyond files recovery forensics", Black Hat USA 2011, 2011, Sec. 8. (Online) Available: http://www.owade.org/

[6] Murilo Tito Pereira, Forensic analysis of the Firefox 3 Internet history and recovery of deleted SQLite records, Digital Investigation, Volume 5, Issues 3–4, March 2009, p. 93-103, (Online) Available: http://www.sciencedirect.com/science/article/pii/S1742287609000048

[7] Oh, Junghoon., et. al., Advanced evidence collection and analysis of web browser activity. Digital Investigation, August 2011, p.S62-S70. (Online) Available: www.dfrws.org/2011/proceedings/12-344.pdf

[8] NSSLabs, Analysis Briefing, "Did Google pull a fast one on Firefox and Safari users?", February 2012, (Online) Available: http://www.nsslabs.com/research/

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)