Search Engine Poisoning and Its Prevalence in Modern Search Engines

Submitted in partial fulfilment of the requirements of the degree Master of Sciences of Rhodes University

Pieter Blaauw

14 December 2012

GRAHAMSTOWN, SOUTH AFRICA

Abstract

The prevalence of Search Engine Poisoning in trending topics and popular search terms on the web within search engines is investigated. Search Engine Poisoning is the act of manipulating search engines in order to display search results from websites infected with malware.

Research done between February and August 2012, using both manual and automated techniques, shows us how easily the criminal element manages to insert malicious content into web pages related to popular search terms within search engines.

In order to provide the reader with a clear overview and understanding of the motives and the methods of the operators of Search Engine Poisoning campaigns, an in-depth review of automated and semi-automated web exploit kits is done, as well as looking into the motives for running these campaigns.

Three high profile case studies are examined, and the various Search Engine Poisoning campaigns associated with these case studies are discussed in detail to the reader.

From February to August 2012, data was collected from the top trending topics on Google's search engine along with the top listed sites related to these topics, and then passed through various automated tools to discover if these results have been infiltrated by the operators of Search Engine Poisoning campaings, and the results of these automated scans are then discussed in detail.

During the research period, manual searching for Search Engine Poisoning campaigns was also done, using high profile news events and popular search terms. These results are analysed in detail to determine the methods of attack, the purpose of the attack and the parties behind it.

Acknowledgements

This thesis would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, my wife Lisinda Blaauw for the encouragement and support during the two years this study has taken.

Dr. Barry Irwin for his support and encouragement during the coursework and thesis periods of our degree.

Professor George Wells for his unfailing support as my thesis supervisor.

Etienne Stalmans for his help with programming issues related to the data gathering done in this thesis.

Thank you to Adeline Levescot for proof reading and highlighting grammar and spelling issues in this document, all remaining faults are my own.

Table of Contents

Table of Figures

1.Introduction

According to Reuters [1], there were over two billion users on the internet in 2010. This is nearly a third of the world population. By 2012 New Media[1] reports that this number increased to 2.2 billion, representing 37% of the world's population. Google announced in 2008[2] that it had processed over one trillion links. Helping such a vast amount of users look for the right content in the ever-expanding internet are the search engines.

Users rely on search engines to provide them with an easy method of finding the information they require on the internet. As of the writing of this document, there are currently two large players in the search engine space: Google's search engine, and Microsoft Bing. While there are other, smaller search engines, this document will focus on these two large entities due to their market share. Internet users rely on them to provide them with information relevant to their search, even if the search terms are vague and sometimes slightly incorrect (e.g. a spelling mistake in the search term, or words in the incorrect order).

Hotchkiss et al [2] studied the effect of the 'Golden Triangle', an area on a Google search page that attracts the most eye scan activity, and states that "unless your site is listed in the top three sites on the page, your chances of being seen by a searcher is dramatically reduced". It is this attempt to attract users to visit sites that has led to a variety of methods being developed to enhance the chances of users visiting a particular site. Some of these methods involve placing the sites in this Golden Triangle, thus increasing the chance of visibility of the site, and the chance that the user will visit the site. Achieving high rankings is what drives an entire industry, and has also caught the interest of those who have less than honest intentions on the web.

The method of improving the chances of sites appearing in high-ranking positions in a search

[1] http://www.newmediatrendwatch.com/world-overview/34-world-usage-patterns-and-demographics
[2] http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

engine's listing is called Search Engine Optimisation, and has spawned a whole industry devoted to it, as will be discussed later in the paper. Users on the internet are not above abusing the ranking system of search engines, and will alter the ranking on purpose through various different ways. We look at the phenomenon of Google Bombing as such an example in section 2.2.

Part of user abuse (malicious or not) to improve visibility of a site, i.e. be part of the Golden Triangle, has resulted in an increase in search engine manipulation. Search engine manipulation is defined by Imperva[3] as "manipulating search engines to display search results that contain references to malware-delivering websites".

The criminal elements and their attempts to gain entry into this Golden Triangle as described by Hotchkiss et al., has also led to the term 'Search Engine Poisoning' being coined, sometimes referred to as 'Search Engine Manipulation', depending on how successful these attempts are. Thus, the legitimate way of influencing search engines is called Search Engine Optimisation (SEO), while the illegitimate ways of influencing search engines can be called Search Engine Manipulation (SEM), or Search Engine Poisoning (SEP). In our study, we will mostly be using the term Search Engine Poisoning.

There are a multitude of methods to perform Search Engine Poisoning, including taking control of popular websites, using the search engines' "sponsored" links to reference malicious sites which inject HTML code, offering documents that execute code in the background and so on.

These descriptions also indicate the concern felt by information security practitioners over this phenomenon. Delivering malware to users in a corporate and private environment allows the criminal element to abuse the resources of the search engines in various ways, from keylogging to stealing banking and other personal details, to extorting money via fake anti-virus software.

With this in mind, the primary objective of the research in this thesis is then as follows:

[3] http://www.imperva.com/resources/glossary/search_engine_poisoning_sep.html

Investigate the prevelance of Search Engine Poisoning within the top trending topics in modern search engines via both manual and automated methods. Doing this will depend on several secondary research objectives that need to be investigated, as follows:

1. Firstly, to look at and investigate case studies of Search Engine Poisoning in 2011, particularly with regards to news events that made headlines. This was done to establish a baseline for the type of results that we hoped to find in the manual and automated research chapters. Several examples were found and explored in detail in Chapter 4.

2. The second objective of the research was to look for Search Engine Poisoned sites using manual search methods. During this period of research several examples of Search Engine Poisoning were found with relative ease, and investigated in detail. The results were compared to historic data from academic and industry sources, and allowed us to make comparisons to determine behaviour changes in techniques and results, along with proving just how easily these campaigns still catch the average internet user. Manual methods will try to imitate the average user on the internet, and thus look at how the normal user behaves when searching for results on the internet, and look at the results presented to the user, including possible Search Engine Poisoned results.

3. The third part of the research involved retrieving data over a period of seven months and running this collected data through various tools as described in Chapter 3. The collected data was then analysed for trends and the results presented. The automated method will harvest the top URL's within Google's top trends on a daily basis. Since this would be near impossible to do on a manual basis, we will be relying on an automated method to collect and analyse this data.

2.Literature Review

2.1 Introduction

In this chapter is discussed the research published on a number of issues related to Search Engine Poisoning. Studying current industry and academic research methods and findings allows us to position our research in relation to these other studies. Several of these studies take a more narrow approach than ours, focusing on fake anti-virus software, particular methods of web injection, or just ways to combat Search Engine Poisoning.

We start with an overview of a particular form of Search Engine Poisoning in section 2.2, the general manipulation of search engines often referred to as “Google Bombing”. Following in section 2.3 is a discussion around the problems associated with web spam, a different method of influencing search engine rankings. This is then followed by a detailed discussion of prior research into the field of Search Engine Poisoning in section 2.4.

2.2 Google Bombing

In studying Search Engine Poisoning, one cannot ignore the phenomenon of 'Google Bombing'. The New Oxford American Dictionary[4] defines Google Bombing as "the activity of designing Internet links that will bias search engine results so as to create an inaccurate impression of the search target"[5]. Hamilton [3] investigates this phenomenon, including the reasons why this is done, and the possible countermeasures to it.

Hamilton states that Google Bombing is the action of setting many pages or sites to a single link, thus associating the target with a key phrase in Google's pagerank algorithm. This

[4] http://oxforddictionaries.com/definition/english/Google+bomb
[5] http://searchenginewatch.com/article/2062070/Google-and-Google-Bombing-Now-Included-New-Oxford-American-Dictionary

association is deliberately done to disrupt the accuracy of the indexing system. Dean [4] explains that Google's indexing system works by partitioning the document IDs into many sections called shards, and each of these shards are replicated onto multiple servers. Dean goes on to explain that in 2001, Google switched to an in-memory index system, which has since been upgraded by Google in 2010 to a system that incrementally updates the index on a continuous basis. Even with these improvements, people are able to pull off successful Google bombs as we show in Figure 1.

Hamilton goes on to mention the first known Google Bomb, an incident where a search query in 1999 — "more evil than satan himself" — returned the Microsoft homepage as its top result. Several other examples are also mentioned, including a 2003 incident involving the erstwhile president of the United States of America, George Walker Bush. Hamilton explains the difference between Google Bombing and Search Engine Poisoning; being that in Google Bombing no actual page content of the target page is ever manipulated or altered when targeted by a Google Bomb. A detailed timeline of well known Google Bombs is displayed in figure 1, based on the work of Tatum [5] and Buck [6].

McNichol [7] takes a closer look at the incident where the biography of President George Walker Bush was manipulated under the Google search term "miserable failure" to be the number one link on the web. McNichol defines Google bombing; stating it is simply the act of taking advantage of the web-indexing mechanism that led to Google being the top search engine. McNichol also credits Adam Mathes, a computer science major at Stanford University at the time, as having coined the term 'Google bombing'.

McNichol states that ever since 1995 when the first search engines appeared, people have tried to manipulate search results on the web. According to McNichol, what is important is not the number of links, but the popularity of the sites linked to the Google Bomb, and the relative obscurity of the term. Bryan [8] explains the mathematics behind Google's ranking algorithm and how it has progressed and been improved over time. Due to the constand changing environment on the web, the algorithm is also changed on a regular basis to accommodate for these changes.

Bar-Ilan [9] looks at Google Bombing from a time perspective. Much like both Hamilton and Nichol, Ilan confirms that Google Bombing is a technique used to actively manipulate search

results. Ilan states that an entire industry has emerged that focuses on Search Engine Optimisation, and finds that bloggers are heavily involved in Google Bombs. Ilan's paper focuses on Google Bombs that were done for humour, ego, and justice, and specifically excludes Google Bombs that were done for financial gain.

2.3 Web Spam

Hayati and Potdar [10] touch on Search Engine Poisoning in a paper published in 2009 called "Spammer and Hacker, Two Old Friends". They define web spam as web pages that are created to manipulate search engines or deceive users, and deliberately trick search engines into offering misleading search results and then serving the malicious results to unsuspecting visitors to the pages. The paper goes on to explore the relationship between hackers and spammers, and the reasoning behind why the relationship is interdependent.

The paper is a problem statement, and thus does not give conclusive methods for preventing Search Engine Poisoning, but does explore the reasons for both spammers and hackers using the technique.

Gyongyi and Garcia-Molina [11] describe two different techniques associated with web spam. The first includes boosting techniques, i.e. “methods through which one seeks to achieve high relevance and/or importance for some pages”, and the second category of techniques includes methods for hiding the technique from the human eye. These various techniques can include any of the following methods: Body spam, title spam, meta tag spam, anchor text spam, repetition of specific terms, dumping of large numbers of unrelated terms, weaving of spam terms into content, and phrase stitching, which spammers use to create content quickly.

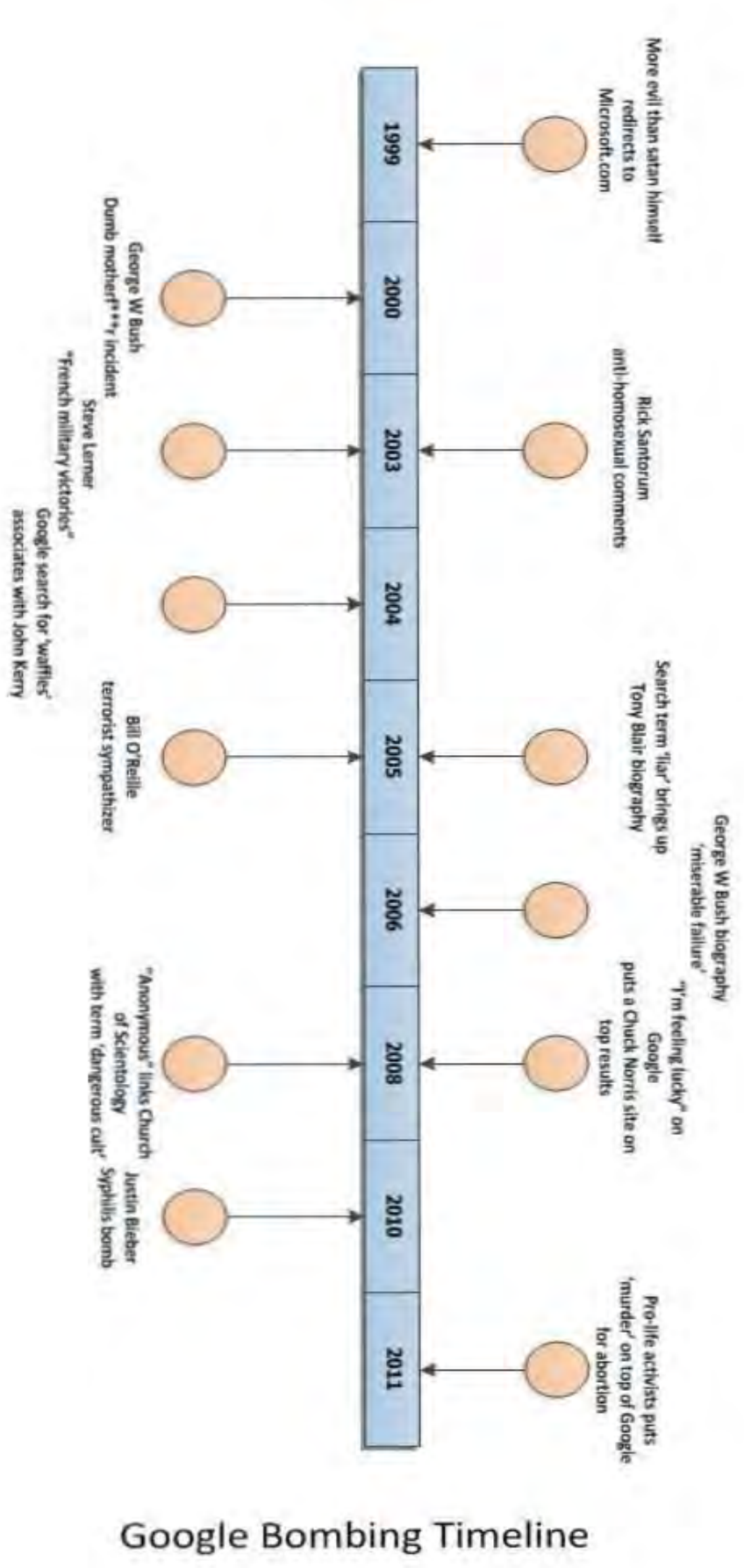

Figure 1: A timeline of important Google Bombs (based on the work of Tatum [5] and Buck [6])

Another technique is to generate blog comments with spam, embedding links in the comments and loading the comments with keywords. Mishne [12] does an analysis of these methods used to spam comments in blogs, as well as the methods that web authors use to prevent it. These methods to prevent spam include registration of users wishing to post comments, preventing HTML in comments or requiring comment posters to solve a *captcha*. Yan [13] defines a *captcha* as "Completely Automated Turing Test to Tell Computers and Humans Apart".

Where Mishne did a comparison of methods, Thomason [14] does an analysis of the different vectors used for web spam. The first is the spammer blog post, a post created by the owner of a website or compromised account with the correct credentials on a blog. The second is the comment spam as described by Mishne. The third vector is the trackback spam. This is a server to server notification of one post that references another, via a fixed API and the specification makes no mention of verification, thus almost any URL can be entered in to a trackback comment. This allows for the linking of possibly malicious URLs without through the trackback method.

2.4 Search Engine Poisoning

Caverlee et al. [15] mention Search Engine Manipulation, also referred to as Search Engine Poisoning in a 2006 paper on Countering Web Spam Using Link-Based Analysis. The authors had already noticed that a considerable amount of malicious website spamming is focused on manipulating the ranking algorithms that drive search engines. These pages then create links to spam pages, and as technology has evolved, these spam links lead to the possibility of malicious exploitation of the end user.

Wang et al. [16] explored investigating websites that exploit unpatched Windows XP systems. The investigation introduces the concept of Automated Web Patrol, an attempt which aims to significantly reduce the cost of monitoring malicious web sites in an attempt to protect internet users. The research goes on to study a number of websites through the use of several versions of Microsoft Windows XP, reporting on the number of exploits found while crawling the web through the automated web crawling mechanism. Wang et al. further find that the major exploit providers own many of the top level sites visited during the malware

study. One owner of three of the top level sites was found to have a redirect relationship with 61 other exploit sites.

Finally Wang et al. look at three groups of interest. Firstly, a group of sites that consists of sites with front-ends that appear to be normal e-commerce sites, that redirect to sites that serve exploits via URLs, which in turn are hosted by five advertising companies. The second group that Wang et al. look at is URLs serving screen savers. Wang et al. claim that many freeware programs bundle spyware in to their installation files, as well as freeware sites actually installing spyware through exploits. The third group of sites consists of malicious search sites, and Wang et al. found more than twenty search sites in their list of exploit URLs. The researchers also looked at what percentage of the exploit URLs they gathered was found in popular search engines such as Yahoo! and Google. The results proved that over 13% of the exploit URLs were served in results by the search engines, which, as stated by the researchers, "is not a trivial percentage". Our research will explore this third phenomenon more deeply.

John et al. [17] developed a system called deSEO for identifying and protecting against Search Engine Poisoning attacks. As mentioned in Wang et al., John et al. found the same attack patterns. These attacks compromise legitimate web servers and generate a large number of fake web pages. John et al. find that by using the ranking algorithms used by popular search engines through Search Engine Optimisation techniques, the attackers are able to poison the search result for popular terms or trends, and thus send users to links with malicious content. The use of the search engines gives the attacker a low cost and legitimate looking appearance. Since the findings of John et al. are the same as Wang et al. in that the sites are mostly based on compromised web servers, the attacks also ride on the reputation of the compromised server.

John et al.'s [17] study suggests that there are two important requirements for a Search Engine Poisoning attack to work: The use of multiple trending keywords, and generation of relevant content across a large number of pages. John et al. also explore cloaking techniques used by attackers to hide the poisoned search engine content from search engine crawlers. The researchers find that there are also three major components in a Search Engine Poisoning attack: The compromised web servers, the redirection servers, and the exploit servers.

During the research done by John et al. they also found that most of the keyphrases used by the Search Engine Poisoning sites are obtained from Google's 'hot trends' and Bing's 'related searches'. John et al. conclude why the attacks are successful: The attackers generate pages with relevant content, target multiple search keywords to increase coverage, and create dense link structures to boost the page ranking.

Leontiadis et al. [18] investigated the use and manipulation of search engines in the promotion of the unauthorised sale of prescription drugs. The researchers constructed a representative list of drug-related queries and gathered search results for a nine month period, focusing their research on a variant of Search Engine Manipulation involving compromised web servers, instead of the email spam that these online retailers have relied on for a long time. The research goes on to show that Search Engine Manipulation is becoming the attack of choice for online criminal operations. Leontiadis et al. also go on to explore the techniques of 'link stuffing' and how criminals use certain 'cloaking' measures to hide the fact that the site has been compromised from the valid owners, as also explored by John et al.

Leontiadis et al. also note an important difference in their research, which sets it apart from the research performed in this thesis. Very often the victims are not victims of 'drive-by downloads' but rather users actively looking for illegal pharmaceutical products.

In their detailed paper, Howard and Komili [19] state that malware distribution through Search Engine Poisoning is “beautiful in its simplicity”. They detail the use of Search Engine Optimisation kits (usually written in PHP) to create web pages filled with keywords that are topical at the time, and, as such, will be crawled by the web search engines. The SEO kits rely on on using content that users are actively seeking on the internet. While the method of distribution of malware works without being stopped, there is little need for the malware authors and distributors to change the method. The paper focuses on how SEO is used in a negative context to improve the rankings of sites serving malware, how the malware authors use the methods often published by search engines on how to improve ranking in order to improve their own rankings, and the many ways that users are redirected once they have landed at an SEO poisoned website.

Methods of hiding SEO attacks within legitimate sites are shown (the researchers found that all but one of the sites they investigated were hosted within legitimate sites), as well as how

links are posted on legitimate sites via user comments and the like to increase the ranking of the page. Howard and Komili show the relationship between hidden SEO kits and compromised content management systems, e.g. Joomla!, WordPress, phpBB and MediaWiki. Analysis of the SEO kits also show how content is dynamically generated, using search engines to source the relevant text for page content. Howard and Komili state that during their research they have seen both the Google and Bing search engines used by the SEO kits, and show the steps of the SEO pages as they are generated by the kits.

Having seen how malware can be distributed through SEO poisoning in the Howard and Komili paper, Caballero et al. [20] look at commoditisation of malware distribution — in other words, turning the distribution of malware into a business, as discussed in Chapter 5.

In a series of articles on the ZDnet website [21] [22] [23], Ed Bott looks at the phenomenon of Search Engine Poisoning. In the first article [21], the author shows a fairly common infection vector for malware distribution. The author describes what happens when the software is downloaded and installed, as well as how to identify the software once it has been installed on a PC. Bott then examines, in an August 8th, 2011 article [22], how the Bing search engine served malware to the user when it did a search for specific phrases that were poisoned. As found in previous papers by Howard and Komili, and Leontiadis et al., Bott finds there is a strong connection between the redirection of the malware or poisoned site and online pharmaceutical sites, as well as other less desirable content being served. Bott also does an analysis of the downloaded file and provides interesting feedback, which we will look at later in this thesis.

Bott further explores [23] the prevalence of Search Engine Poisoning by searching for several key phrases and following poisoned links to sites that serve malware under the pretence of being legitimate software downloads. Bott identifies that the SEO kit used by the malware writers uses a polymorphic engine for the files, which was not detected at the time of writing of the research article. Newsome et al. [24] define polymorphism as a way “through which a program may encode and re-encode itself into successive, different byte strings, enabling production of changing payloads”.

Research into SEO poisoning is not limited to academic institutions. Blue Coat, a company that specialises in web security and network optimisation released several SEO findings in

their 2011 Web Security Report [25]. Blue Coat states that during 2010, the methods of delivering malware on the web migrated to trusted domains, through hacking or compromised credentials. This is done specifically to rely on the trust and reputation that the hacked domain has. Blue Coat also finds that what makes SEO poisoning such an effective method of crime and malware delivery is human interest and the fate of others, and that the defences that monitor websites are often blind to the threats hidden in legitimate sites.

Blue Coat [26] confirmed what Bott, Howard and Komili all found, that fake anti-virus accounts for 60% of malware found on the domains that use SEO poisoning. Blue Coat researchers found that SEO kits place parts of their infrastructure all over hacked and reputable domains and websites to avoid detection. This challenges web protection measures that rely on reputation since trusted sites are now part of a method referred to as 'dynamic link chains'. Link farms then continue to hide in trusted domains and poison search engine results.

Larsen [27], a researcher at Blue Coat, also reports via an explanatory graphic in Figure 2 that search engine poisoning was the most prevalent method of delivering web users to sites that deliver malware.

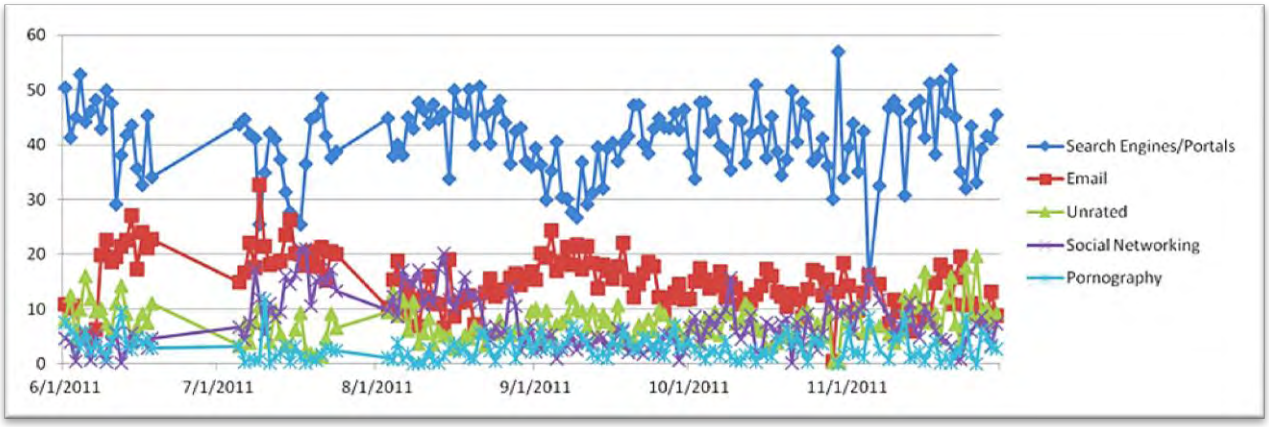

Figure 2: The origin of malware attacks, Larsen [27]

The blue line represents the average percentage of where malware attacks originated. As can be seen from this graphic, Larsen and the researchers at Blue Coat found that nearly forty percent of all malware attacks orginate from Search Engine Poisoning attacks, with e-mail, social networking, and other unrated origins accounting for the rest.

An interesting part of this graph shows how little impact pornography has on the spread of malware and malware networks. This can be attributed to the fact that pornographic websites already have an established source of revenue, and thus don't want to endanger that source by being associated with the spread of malware.

While Bott investigated some of the malware distributed through SEO poisoning, touching on the subject of fake anti-virus, Rajab et al. [28] analysed fake anti-virus distributions in detail. This trend is confirmed in a white paper published by the Lionic Corporation [29] in March 2011. They find that the bulk of the poisoned pages found during their research were used to send users to fake anti-virus pages, trying to persuade users to install the fake anti-virus software. We look at this paper purely because of some of the statistical value that it provides in SEO poisoning and the impact it has on the fake anti-virus industry.

Firstly, Rajab et al. find that during their research period of 13 months, the percentage of infected domains offering fake anti-virus increased from 3% to 15%. The researchers also found that in March 2009 the ratio of fake anti-virus domains to landing pages (SEO pages) was approximately 96:1. Another finding by Rajab et al. is that distributors of fake anti-virus are more successful at targeting domains and trending keywords than distributors of other types of malware. Finally, Rajab et al show that fake anti-virus malware at the time of writing their paper accounted for 15% of all the malware they identified.

Fisher [30] reported on an SEO campaign that appeared in October 2011 via the Microsoft Bing search engine. Users searching for Adobe Flash downloads are led via the techniques discussed earlier to a compromised website, where a trojan masking as the executable file users are looking for is downloaded. The executable is a piece of malware known as ZeroAccess Trojan. This confirms what the Blue Coat report and Bott found during their research, and proves that SEO poisoning is still a very effective method for delivering malware to unsuspecting users.

Clay [31] reported that McAfee, in their 2012 annual report on the most dangerous web celebrities, ranked Emma Watson as the most dangerous celebrity to search for online, with a

12% chance of getting infected with malware. Watson[6] gained fame as the female lead in the Harry Potter series of movies, playing the character of 'Hermione Granger'. In the previous year, model Heidi Klum topped the annual McAfee list..

Leydon [32] reported that a study by Sophos Labs found that the Microsoft Bing search engine was the most heavily poisoned search engine compared to others. The article reported that 65% of the poisoned results originated from the Bing search engine, while only 30% came from Google. The article also reports that 92% of the blocked results come as a result of image searches, and only 8% from text searches. Howard Fraser from Sophos Labs reports that search engines continuously play a cat and mouse game, and that the search engines are not always successful.

2.5 Summary

From the research papers to industry reports studied in this literature review, we can see that one thing is obvious: Search Engine Poisoning, in one form or another, is a phenomenon that has been with the search engine industry since the beginning. Hamilton [3] showed us that Google as a search engine was only a year old before the first type of search record manipulation took place, while Larsen [27] has shown us that it is still the primary way to deliver malware to a victim's computer.

Commercial research from McAfee [31], Sophos Labs [19] and BlueCoat [27] [26], as well as academic research have showed us the relationship between the criminal elements on the internet, and their need to manipulate the results of search engines. Research from Fisher [30] as well as Bott [21] [22] [23] shows the methods used by the various elements, and the end-goal of these elements, be it malware installations, fake anti-virus or even more harmful payloads.

In the next chapter the research approach in this thesis is discussed, including the case studies that are studied, the manual and automated data gathering methods and the subsequent data analysis that will be performed.

[6] http://en.wikipedia.org/wiki/Emma_watson

3.Research Approach

3.1 Introduction

While the research in the literature review focused on methods of Search Engine Poisoning as done by Howard and Komili [19], the fake anti-virus industry and the link to Search Engine Poisoning studied by the Lionic Corporation [29], or the pushing of prescription drugs via online sales done by Leontiadis et al. [18] among others, the research conducted in this thesis will focus on three distinctive areas in a three part discussion.

Firstly, case studies relevant to the thesis will be studied in Chapter 4 in order to explain the phenomenon of Search Engine Poisoning and why it is still so effective today. In this section, the deaths of Osama Bin Laden and Amy Winehouse are studied, as well as a smaller SEP campaign relating to a different news making event, where a teacher named Ileana Tacconelli was fired from her job, and this made news to such an extent that it caught the attention of the criminal elements that operate the SEP campaigns.

The second part of our research in Chapter 6 will focus on the human element, searching the internet manually, looking at terms which have been classified as dangerous, and seeing what results can be gathered in such a way. These results are then run through five different virtual machines and the results are collected and analysed and studied to see the effects of normal Search Engine Poisoning on the normal internet user.

The third part of the research approach in Chapter 7 focuses on automated searching, gathering data in an automated fashion, and logging and storing the data for analysis via various tools available. Data is collected via a virtual private server that was rented in a data centre, on a daily basis, and this data is then processed through automated tools, and the results analysed. The data that has been collected over the time period is then analysed for patterns, and interrogated for results and conclusions drawn from said data.

3.2 Case Studies

During 2011 there was a host of case studies that have been exploited in Search Engine Poisoning attacks. These include, but are not limited to, the deaths of Osama bin Laden, Amy Winehouse, Steve Jobs and Muammar Gaddafi. All of these high profile deaths were followed by some form of Search Engine Poisoning attack.

The aim of the research is to look into these high profile deaths, analyse the time frame from incident to the first reported case of search engine poisoning, and to look at which attack avenue was used in the attack, be it social media, image poisoning or a normal Search Engine Poisoning attack via web links contaminated with malware. Chapter 4 will also show graphs of the searches for information on the case studies, and these give a clear indication of the search volume for information around the news event, and why these events became the focus for Search Engine Poison campaign operators.

A good example is the Kaspersky Blog post that followed the Osama bin Laden death[7] and the subsequent images that were search engine poisoned. The images redirected to rogue anti-virus sites that offered a well known fake anti-virus program called 'Best Antivirus 2011'. Each of the four celebrity deaths mentioned had a Search Engine Poisoning campaign attached to it, thus allowing us a good base to search and extract data from.

Not only celebrity deaths will be looked at, but also other news events which have had a search engine poisoning campaign attached to them. Each of the case studies will show a different method of manipulating search engines with a different aim in the end, though one could come to the conclusion that the ultimate aim of all the search engine manipulators are monetary benefit in some form or another.

3.3 Manual Searching

The next avenue to look at during the research is manual searching. We aim to look at the popular ways that search engine poisoning have taken place in the past and then simply follow the same methods to look at finding pages that have been the victims of search engine

[7] http://www.securelist.com/en/blog/6202/Blackhat_SEO_and_Osama_Bin_Laden_s_death

poisoning.

McAfee named Heidi Klum as the most dangerous celebrity on the web for 2011 [33] , and Hope Dworaczyk, a former Playboy Playmate was awarded the title of 'Miss Malware' [34] at the Defcon 18 conference in 2010. This focus on celebrities being used in Search Engine Poisoning campaigns is used as the starting point for the manual searching that is used in this thesis. Part of the research will be to look at campaigns launched by malicious SEO operators, and impersonating the average user while searching the web for exactly what McAfee has identified, and hoping that it leads us to a poisoned search result, thus allowing us to become 'victims' of the attack.

This methodology differs from the automated searching discussed in the next part in that human behaviour can be very different to the behaviour programmed into a piece of software or a script running every few hours, even if a low interaction honey client is used (as discussed in section 3.4.3).

Part of the research methodology will also be to look at how good Google's 'safe search' function is when looking for images using Google's Google Images search function. Larsen [26] has reported that the 'safe search' option of Google's algorithm is fairly easily bypassed, and thus we will look at how easy it is for Search Engine Poisoning campaign operators to bypass this feature.

3.4 Automated Searching

A substantial part of the searching for search engine poisoned web pages was done using automated searching. The idea for the automated part of the research came from Moore et al [35], who noted that on 24 February 2011 Google changed rankings in its search engine algorithm to eradicate low-quality results. Moore defines low quality sites as "low-value add for users, copy content from other websites or sites that are just not very useful". Moore found that a lot of search engine poisoned sites fall under this. Our research commences after these changes were carried out by Google, and we should see if the changes have been effective or not.

The first part of this comprises historical data that gathered for Google Trends for the last three years, collected into a database and analysed for content and links. These trends should give us more insight into the history of certain trends, and allow us to collate data collected with Search Engine Poisoning campaigns in the past.

Matching data collected with historical campaigns allows us to look at possible response times to news by black hat SEO operators, and allows us to view what malware types have been offered or pushed onto unsuspecting internet users. This is especially interesting since the takedown of the Russian payment processor Chronopay, and its links to rogue anti-virus products[8]. We do this because of the link that the fake anti-virus products vendors have with the Search Engine Poisoning campaign operators as pointed out earlier by both the Lionic Corporation [29] and Rajab et el [28].

The second part of the automated searching comprises a large part of this research paper. The process of collection will simply be gathering the Google Trends on a daily basis, following each trend, gathering the top 10 URLs for the trend, and logging this into a database. This data gathering period will happen for seven months, from 1 February 2012 to 1 September 2012 during the thesis writing period, giving us data for at least seven months of 2012.

The URLs gathered will then be run through various engines and API's in order to perform automated testing for any malware, cross-site scripting, malicious javascript, or silent malware downloads.

3.4.1 JSunpack

The first tool that will be used for automated URL processing is 'jsunpack'. The 'jsunpack' [36] tool is a Python script for detecting and analysing malicious JavaScript embedded in websites. It emulates web browser functionality when visiting a web site and tries to detect browser and browser plug-in vulnerabilities.

Using the 'jsunpack'[9] tool each URL will then be tested for malicious javascript via

[8] http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/reports/rpt_a-look-back-at-2011_information-is-currency.pdf
[9] http://code.google.com/p/jsunpack-n/

automated scripting, and the result logged to a database for further analysis. The tool also allows us to interrogate Adobe .pdf documents which may contain malicious code embedded in the document.

3.4.2 VirusTotal

The second service used in this thesis for testing the status of a web page being retrieved from the Google top trends list is the VirusTotal[10] service.

As per the definition on their homepage, "VirusTotal is a free service that analyses suspicious files and URLs and facilitates the quick detection of viruses, worms, trojans and all kinds of malware".

The VirusTotal API allows us to run the daily collected URLs through a limited API, and query the status of the URLs at a later stage. VirusTotal in return collects data from twenty-six different web security providers and forty-one anti-virus providers allowing us to get a holistic view of the status of submitted websites without the need to interface with multiple vendors and databases. While the list is by no means complete, it allows us to get enough information to warrant further investigation into an identified URL or file.

3.4.3 Thug

The third automated method for testing for malicious sites through the Google Trends data gathering will be Thug [37]. Thug is a low interaction honey client that will be used to analyse sites that are collected via both automated data collection and manual searches. Qassrawi and Zhang [38] define a honey client as "an active honeypot that mimics, either manually or automatically, the normal series of steps a regular user would make when visiting various websites".

Based on the research done by Seifert et al. [39], it was decided to use a low intensity honey client instead of a high interaction honey client to analyse the data gathered, as high

[10] https://www.virustotal.com

interaction honey clients are time and resource consuming. Seifert et al. also point to the fact that high interaction honey clients also look at other means of attack, while the research presented here focuses on search engine poisoning attacks via malicious or compromised web sites; thus the need for such a high interaction honey client would be excessive and produce too much non-relevant data.

A detailed diagram and description of the test environment used is presented in section 6.1.1.

3.5 Data Analysis

Once a malicious site has been found that was created and indexed in Google through Search Engine Poisoning, the next step in the research is to analyse the content of the malicious website. This will be done using several virtual machines, each containing a different version of Microsoft Windows (e.g. Windows XP Service Pack 2 versus Windows XP Service Pack 3), and in addition containing several patch levels and different versions of Internet Explorer (e.g. Internet Explorer 6 versus Internet Explorer 8).

While visiting the malicious URLs with the virtual machines, analysis of software downloaded either covertly or willingly will be done, as well as looking at the effect the website might have had on the machine.

By analysing the software downloaded, statistics can be gathered on the type of software that was installed, as well as classification of the possible malicious software found. Analysis of these statistics and detail can also provide insight into the origins of a campaign, and the objective of the campaign operators, be it fake anti-virus advertisement, compromising a computer to become a botnet client, or installing malware or ransomware.

3.6 Summary

While the research methodology is by no means unique, previous research into Search Engine Poisoning has focused on specific areas such as the fake anti-virus industry, and did not use data specifically collected from Google's top trends over an extended period of time, nor

emulate normal user behaviour on the internet.

The methodology used in this thesis thus differs as we will be gathering the top Google trending data over an extended time period and analysing it. In this way, we aim to find out how effective the campaigns by the Search Engine Poisoning operators are, and if they are indeed managing to get into the Google trending list through both manual and automated search methods, as previously defined on page 3.

Lastly, we will also see how effective Google and Bing are in detecting and preventing Search Engine Poisoning campaigns, both in its Google trending data, and in normal searches by users looking for specific topics on the internet.

4.Case Studies

4.1 Introduction

Durkin [40] studies the human fascination with death and dying in the paper 'Death, Dying and the Dead in popular culture'. The paper explores our preoccupation with death in television, cinema, music, print media, recreation, jokes, and other aspects of popular culture. Durkin describes the term 'post self' and how it is especially true for celebrities. The term references the ability of the person to influence, and the reputation the celebrity has after their death. Durkin describes it thus: "the post self constitutes a form of symbolic immortality, whereby the meaning of a person can continue after he or she has died".

This virtual immortality through meaning can be seen in true effect when we look at the death of Osama bin Laden, and the meaning that this event has had to the world. This will be explored further in section 4.2. During 2011 there were several celebrity deaths, and their 'post self' was even seen on the internet, when people were looking for as much information as possible surrounding the death of the celebrities — what caused it and how did it happen? This 'post self' of the celebrity deaths did not escape the eyes of the Search Engine Poisoning campaign operators.

As previously mentioned, during 2011 there were several concentrated Search Engine Poisoning campaigns focusing on a certain celebrity death, and in this section we will look at those centred around these deaths. A brief biography of each celebrity will be given in order to give the reader a clear understanding of why the celebrity death was important, and more so why the celebrity death was important to the Search Engine Poisoning campaigners.

To avoid focusing just on celebrity deaths and the Search Engine Poisoning campaigns around it, we also look at other smaller campaigns relating to other news events, and the objectives of such a campaign.

Google provides us with two very good tools for studying search history for certain terms or topics, and have collated data going back to 2004. These tools are Google Trends[11] and Google Insights for Search[12]. One of the features that will be prominently used during this chapter is the ability of these services to actively graph the search terms for a time period. This will provide an easy to understand visual explanation of how the search term escalated during the news event, and how people's searches for the event tapered off after the event. It is important to remember that the Search Engine Poisoning kits look for these high ranking search topics and then try to poison the searches. Thus we look at these historical events and their search rankings, as well as the Search Engine Poisoning campaigns that followed.

It should be noted that the other major search engines on the internet, namely Microsoft's Bing and Yahoo! Search do not offer the ability to study and graph the major search terms from a historical perspective. Yahoo! does have a service called Yahoo! Buzz[13], but it does not allow for the comprehensive historical search that Google does at the time this thesis was written.

In the following section we will look at the deaths of terrorist leader Osama bin Laden and singer Amy Winehouse, and the controversy surrounding a school teacher named Ileana Tacconelli, as well as the associated Search Engine Poisoning campaigns that followed these news events. Graphs that show the rise and fall of the search terms for the case studies will be shown, as well as for associated terms that users on the internet searched for.

4.2 Osama bin Laden

Osama bin Laden rose to public prominence as the leader of the terrorist organisation Al-Qaeda. Born in July 1957, he was the son of a construction magnate, and went to Afghanistan in 1979 during the Soviet invasion of Afghanistan. After the Soviet withdrawal, Bin Laden moved to Sudan and started training Al-Qaeda militants [41].

[11] http://www.google.com/trends/
[12] http://www.google.com/insights/search/
[13] http://buzzlog.yahoo.com/

In this position, bin Laden masterminded the attacks on two American embassies in Africa on August 7, 1998: One in Dar es Salaam in Tanzania, and one in Nairobi, Kenya. These attacks were followed by a suicide attack on the United States Navy destroyer the USS Cole, while it was berthed in the port of Aden in Yemen [42]. Osama bin Laden then masterminded and was responsible for the attack on the World Trade Center on September 11, 2001 [43].

To understand why this attack, and the subsequent hunt for and eventual finding of Osama bin Laden was of such interest to the world and why it was chosen as one of the case studies for this thesis, we need to look at some numbers. Bock [44] reports that the September 11, 2001 attacks led to the largest loss of life in a single event since the Pearl Harbour attack in World War II. A total of 2,977 American civilians and 19 terrorists lost their lives in the attacks.

Bock also reports that before the World Trade Center attacks, the largest loss from a single event covered by United States insurance companies was Hurricane Andrew. The claims associated with the World Trade Center and accompanying events exceeded fifty billion US dollars.

These statistics show the enormity of the impact that this attack had, and as such, why the capture of Osama bin Laden became one of the biggest targets for the United States of America. The Federal Bureau of Investigation had Osama bin Laden on their top ten terrorist wanted list, offering a reward of USD 25,000,000[14].

On May 2, 2011, the president of the United States, Barack Obama, authorised a raid in Pakistan by United States Navy Seals [45] which ultimately led to the capture and killing of Osama bin Laden. As president Obama announced to the world that Osama bin Laden had been killed, the news went viral on television and news networks worldwide. This obviously also included the internet.

[14] http://www.fbi.gov/wanted/topten/usama-bin-laden

4.2.1 Google Trends and Insights

As the news of Osama bin Laden's death spread throughout the world, people searched on the internet for news: Confirmation that it had really happened, and as much information as possible regarding the circumstances surrounding his death. When looking at the Google Trends data for the search term 'osama bin laden', a marked increase in searches can be seen following the news of his death.

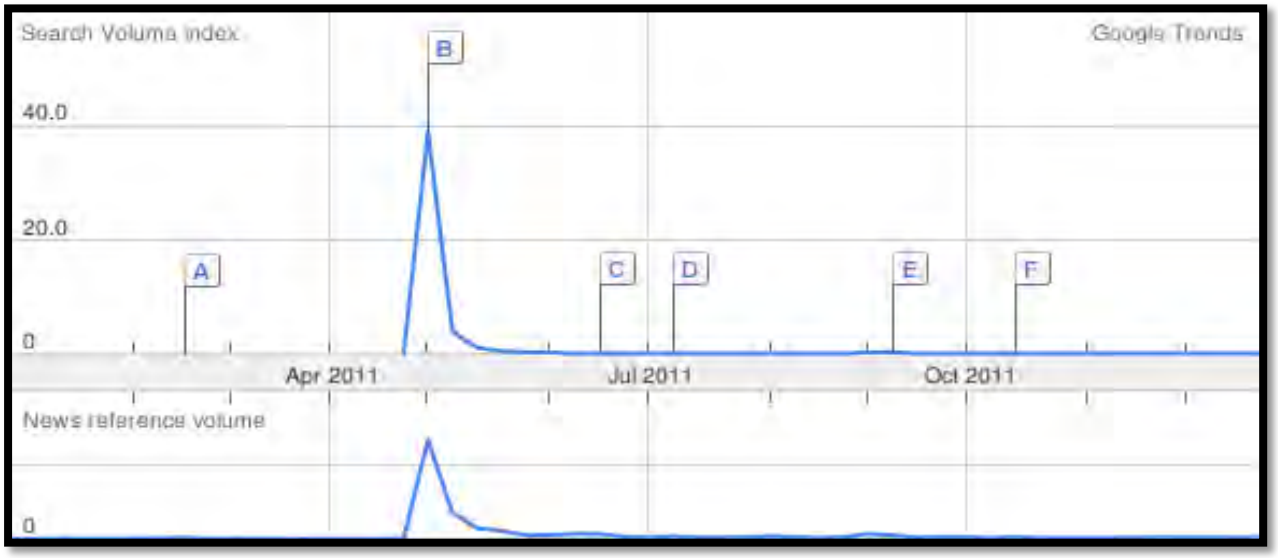

Figure 3: Google Trends showing the sudden rise in searches for 'Osama bin Laden'

The letters in the graphic are Google links to articles, and can not be eliminated from the picture easily. In order to have a closer look (without article links), we look towards Google Insights for Search.

Figure 4: Google Insights for Search showing the sudden rise in searches for 'Osama bin Laden'

In Figure 4 it can be seen that there is a massive spike in the amount of searches for 'osama bin laden' during May (the time of his death), and a minor rise during mid September 2011 when more news regarding the death of Osama bin Laden was released. It needs to be noted that all of the graphs presented in this thesis are historical datasets that Google allows to query, and this data was not being collected realtime for this thesis.

4.2.2 The Search Engine Poisoning Campaign

This increase in searches for news about the death of Osama bin Laden did not escape the sights of the Search Engine Poisoning engine operators, and within hours of the news, several information security companies were reporting poisoning campaigns related to the death of Osama bin Laden, as the searches for news were being poisoned.

The Search Engine Poisoning campaigns took on several different manifestations. Some of the campaigns focused on getting the users to spread it themselves via crude methods on social networking sites, while others tried to trick users into installing fake anti-virus software. In the following section a closer examination of the various methods will be done.

Assolini [46] reports on a campaign that inserted fake images of Osama bin Laden's body into Google Images. Once the unsuspecting user clicked on the links, they were taken to fake anti-virus websites. When the user landed on the page, they were offered a copy of a well known fake anti-virus known as "Best Antivirus 2011", as the screenshot in Figure 5 shows.

Another way in which the black hat operators capitalised on the death of Osama bin Laden was via Facebook. Cluley [47] reports that the users received a post on their profile claiming to have a link to a video that contained the last minutes of Osama bin Laden's life, as shown in Figure 6.

Once the user clicked on the link, they were then asked to perform a few simple tasks, before being able to watch the video. Once the user clicked on the video, they were taken to a survey to complete, instead of the video. Cluley reports that the operators of the scam earned money for every survey completed, thus the attraction of capitalising on Osama bin Laden's death.

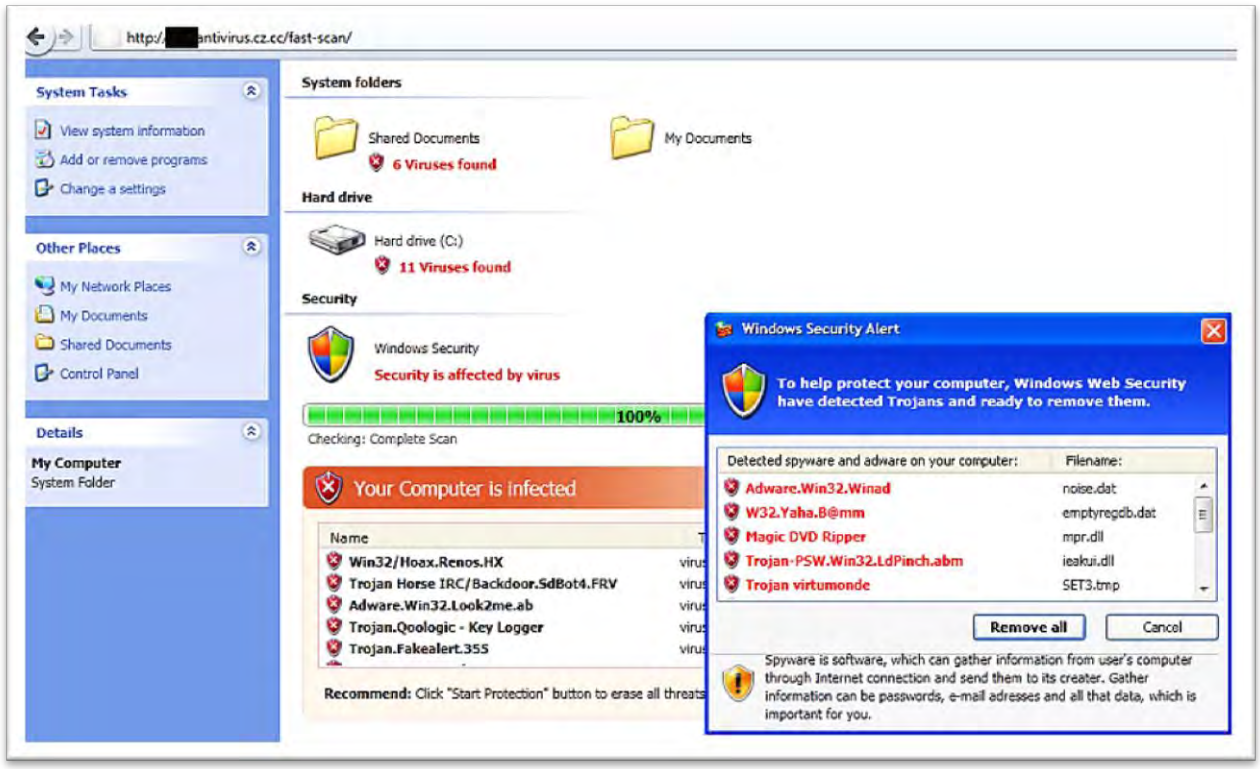

Figure 5. An example of the fake anti-virus program Best Antivirus 2011 [46]

Figure 6. An example of the scam on Facebook used by Search Engine Poisoning operators [47].

Looking at just these two examples, we can see that various scams were rife very soon after the death of Osama bin Laden, all driven by the search for information surrounding the circumstances of his death.

4.2.3 Summary

The finding and subsequent killing of Osama bin Laden after a ten year search was going to make headline news throughout the world, and such a large news event proved to be the perfect springboard for various Search Engine Poisoning campaigns as we have explored in this section.

Various fake news items, fake pictures and Facebook campaigns tried to lure and trick people in to clicking on links or pictures in order to compromise their computers and to try spread the campaigns further. During the study presented, we looked at how the search requests for information regarding bin Laden's death rose, and the different ways how poisoning through links and Facebook campaigns happened. What is clear is that even in his death, Osama bin Laden did live on as described by Durkin [40].

4.3 Amy Winehouse

Amy Winehouse was born Amy Jade Winehouse on September 14, 1983 in London [48]. From an early age she showed musical talent, and in 2003 released her first album 'Frank', followed in 2006 by her second album 'Back to Black'.

Shaw et al. [49] and McRobbie [50] both report on the drug and alcohol problems that Amy Winehouse had, and on 23 July 2011 Amy Winehouse died from blood alcohol poisoning [51]. No sooner had the news of her death reached the mainstream media, than the operators of the Search Engine Poisoning campaigns started poisoning searches for Amy Winehouse.

4.3.1 Google Trends and Insights

As the news of Amy Winehouse's death was reported, people went onto search engines and started searching for news about it. Looking at the Google Trends and Google Insights information for the date range related to Amy Winehouse's death, we can see why this was

again a very good target for the Search Engine Poisoning campaign operators to use.

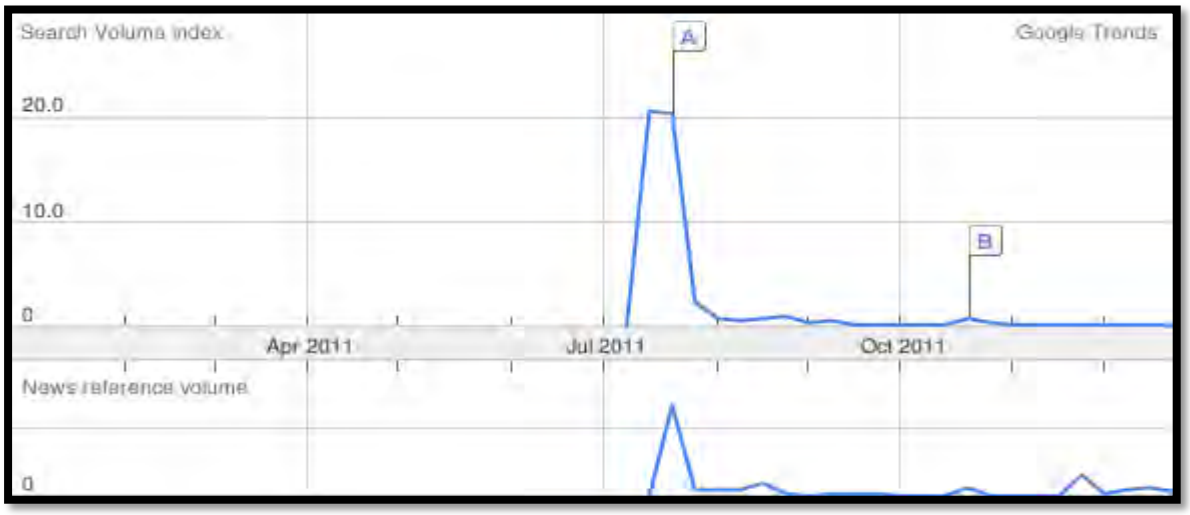

Figure 7: Google Trends showing the sudden rise in searches for 'Amy Winehouse death'

There is a marked increase for searches regarding her death from the time the news was released until after her funeral, as shall be explained with the following graphs (Figure 7 and Figure 8).

As per the Osama Bin Laden searches shown earlier, a marked increase in searches related to the news items can be seen for the time period related to Amy Winehouse's death.

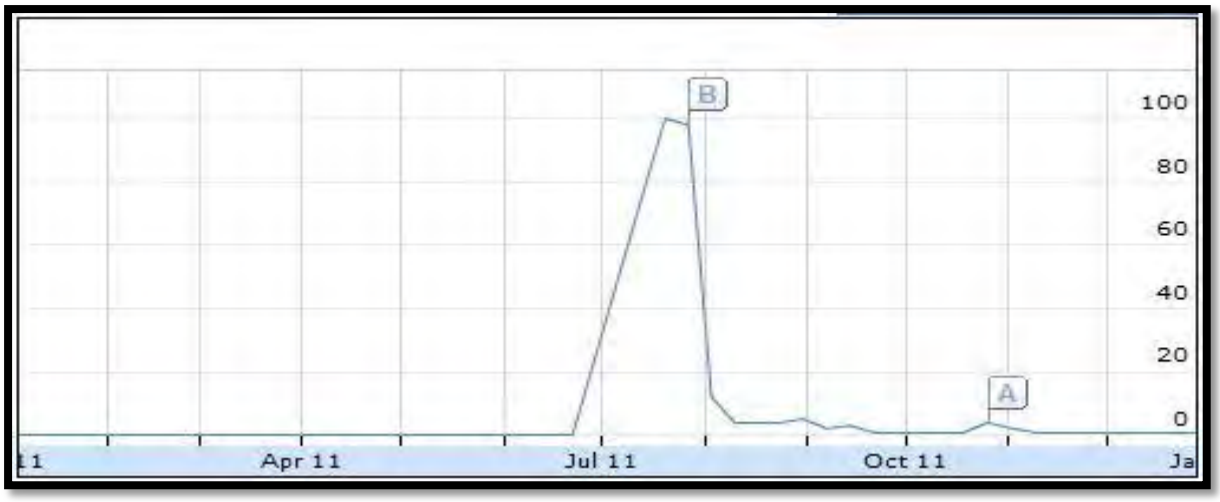

Figure 8: Google Insights for Search showing the sudden rise in searches for 'Amy Winehouse death'

As can be seen from the graphic in Figure 8, there is a marked increase in the searches for

any information regarding the death of Amy Winehouse during the time of her death. Due to the details surrounding her death being vague at the time, the period reflected in the searches is a bit longer than with the Osama Bin Laden search, and the searches only taper off after her funeral.

4.3.2 The Search Engine Poisoning Campaign

Once the death of Amy Winehouse reached the mainstream media, the Search Engine Poisoning campaign operators started their campaigns. The primary medium targeted initially was Facebook, with various scams operating in much the same way as with Osama Bin Laden's death.

Users searching for images of Amy Winehouse were lured into clicking on fake pictures or videos of Amy Winehouse under various headings, including "SHOCKING – Amy Winehouse's Final Minutes" and "Leaked Video!! Amy Winehouse On Crack hours before death" as shown in Figure 9. As Durkin described earlier, users have a fascination with celebrities and death, and thus it was easy for the scammers to lure people into clicking on these pictures or videos.

Figure 9: Examples of the Amy Winehouse Facebook scam [52]

The scam worked the same as with Osama Bin Laden's death, where people are lured into clicking on the picture or video heading and then asked to complete a survey before seeing the video or pictures. The scammers get paid for every survey completed and thus capitalise on people's curiosity.

The SEP campaign was not just limited to social media, with Ragan [53] reporting that email was sent by spammers containing an attachment that claimed to be video of Amy Winehouse moments before her death. The email attachment contained a compressed file, containing the SpyEye Trojan[15].

Lumague [52] also reports of a scam where the search string 'amy winehouse death' entered in to search engines led users to malicious links, which in turn redirected the users to sites hosting malware. The particular malware or malicious file in this instance was a fake anti-virus program[16] (a relationship which we have explored earlier already).

Just as with the death of Osama bin Laden, we see that the operators of the Search Engine Poisoning campaigns use various methods, from social media to normal link poisoning, in order to get users to visit the poisoned link, and thus profit in some way from it.

4.3.3 Summary

The Amy Winehouse case study presented in this section showed us that the world is still obsessed with celebrities, even after they have left us, and they continue in their 'post self' as described by Durkin [40]. With Amy Winehouse's death, it gave the Search Engine Poisoning campaign owners another opportunity to try and lure people in to clicking on fake Facebook pictures or to convince them to click on links or video files that could lead to the compromise of their systems. As with the Osama bin Laden case study, the prevalence of social media is again high in the methods that the operators of these campaigns use.

4.4 Ileana Tacconelli

Pisa [54] reports on the former model Ileana Tacconelli, a teacher at the San Carlo Catholic School in Milan, who caused a storm when racy pictures of her were posted on the internet.

Ileana Tacconelli won a regional beauty contest in Italy [55], crowning her Miss Abruzzo, and entered the national competition for Miss Italy, after which she studied at university to obtain a teaching degree. Having taught at the school for three years, the publication of the

15 http://www.pcworld.com/businesscenter/article/247252/spyeye_malware_borrows_zeus_trick_to_mask_fraud.html

16 http://about-threats.trendmicro.com/Malware.aspx?language=us&name=TROJ_FAKEAV.CLS

photos caused some parents to remove their children from the school, and the story ultimately reached the mainstream news.

4.4.1 Google Trends and Insights

Once news hit of Miss Tacconelli, people immediately went to the search engines to find out who she is and what the fuss was about. Reyes [56] reported that because the articles reported that she was published in rather revealing photos, the image search feature in Google was also quite heavily used. As was reported by Assolini [46] in the Osama bin Laden case, these images can also be used to either redirect a user to a malicious site, or directly infect the visiting computer.

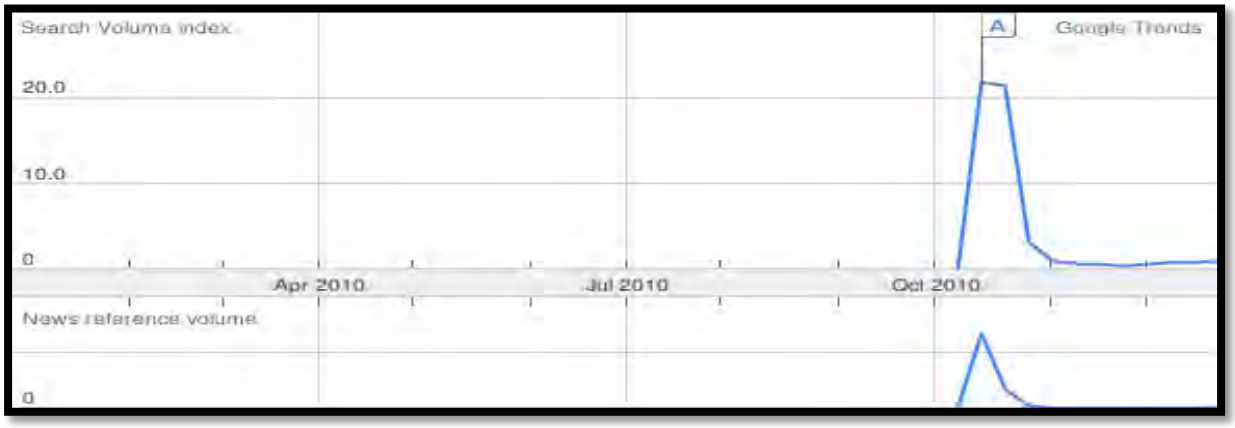

Figure 10: Google Trends showing the sudden rise in searches for 'Ileana Tacconelli'

The news articles about Illeana Tacconelli reached the internet on the 21st of October 2010 and the Google Trends in figure 8 show an immediate search increase for the search term in Figure 10 and Figure 11.

Figure 11: Google Insights for Search showing the sudden rise in searches for 'Ileana Tacconelli'

An interesting observation is that Google Insights for Search reported that the top search terms for Ileana Tacconelli during the time period were actually 'video ileana tacconelli' and 'ileana tacconelli movie'. This differs slightly from the trends we have seen earlier, where the exact search terms used in the SEP campaigns did not differ from the terms in Google Search.

4.4.2 The Search Engine Poisoning Campaign

Reyes reported that one of the interesting facts about the search engine poisoning campaign around Ileana Tacconelli was the relatively small size of the news event. This did not deter the Search Engine Poisoning campaign operators, and soon after the news articles were published, poisoned Google Images started appearing, as shown in Figure 12.

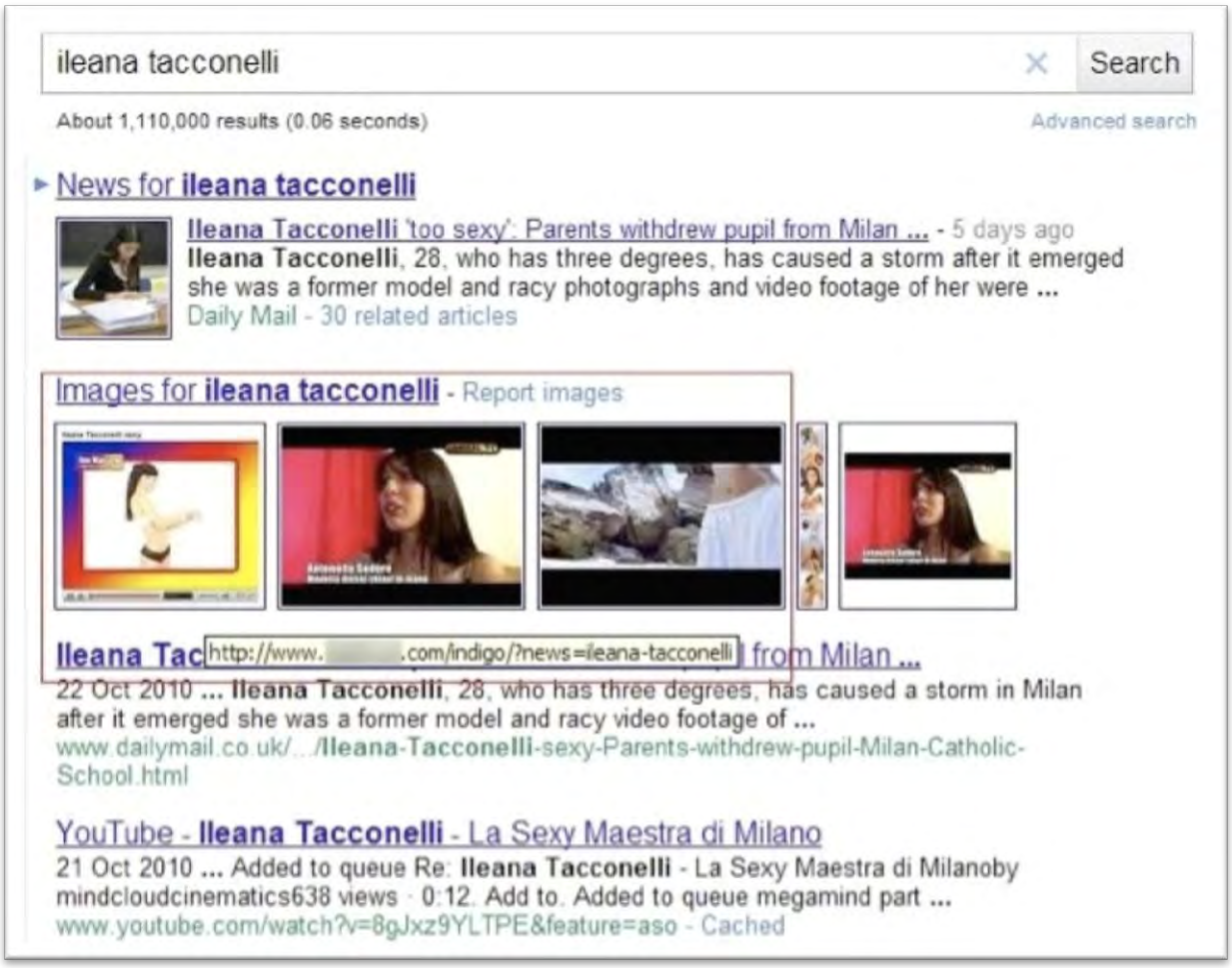

Figure 12: Google Images search showing the poisoned images [56]

Reyes reported that once a user clicked on one of the poisoned images, the user was redirected to several sites and then prompted with a display claiming that their version of Adobe Flash Player was out of date and needed updating as shown in Figure 13.

Figure 13: Screenshot of the fake Adobe Flash Player update [56]

Once downloaded and executed, the file is actually part of the TDSS rootkit. Goodin [57] reports that the TDSS rootkit first appeared in 2008, and is constantly updated. Goodin also states that the rootkit is particularly effective, in that it has a built-in encryption scheme that prevents the monitoring of communication between the infected PC and the control servers.

An interesting difference in this search engine poisoning campaign was the fact that this campaign was aimed purely at gaining more hosts for the botnet, whereas the other case studies in this document were victim to survey scams and installing fake anti-virus software. It is clear that Search Engine Poisoning is used by cyber criminals for various different attacks through various different media including normal internet searches, image searches, and social media.

4.4.3 Summary

As has been shown in this section, the Search Engine Poisoning campaign operators not only rely on the death of a public figure or a celebrity to launch a campaign. The Ileana Tacconelli campaign is a good example of how the operators of these campaigns also focus on sexual

curiosity to lure victims to click on pictures or links. By posting pictures of the subject in a state of undress, it piques the interest of those who might want to see more of the subject and thus lures them to the campaign. In this case, also offering a video was another attempt at luring victims, and obtaining botnet victims as described by Goodin [57].

4.5 Summary

Looking at the three examples that were studied in this chapter, it becomes clear that current news items, celebrity deaths and even the smaller headlines can be targets for those operating Search Engine Poisoning campaigns.

The goals and objectives of those running the campaigns vary slightly; some being survey scams, and others trying to obtain more hosts for a botnet network. It is clear though, that they all operate by exploiting human curiosity — the very curiosity Durkin [40] mentions — for their own profit at the expense of normal internet users.

5.Browser Exploit Packs

5.1 Introduction

Villeneuve states [58] that "Cybercriminals use distribution methods such as spamming and black hat search engine optimization (BHSEO) techniques, often in conjunction with exploit packs that can take advantage of vulnerabilities in popular software in order to distribute malware to unsuspecting Internet users."

According to Sood and Enbody [59] "Browser Exploit Packs thrive by exploiting the browsers' vulnerabilities, and attackers have demonstrated a lot of maturity and expertise in developing their exploits. BEPs are usually used in conjunction with botnets and use drive-by-download attacks to load the malware binary onto the victim's machine."

With this in mind it is important that research be done into how these attacks take place, what tools are being used by criminal elements, and how these tools work, as this has direct influence on our research into Search Engine Poisoning. These are the very tools used in SEP campaigns. Thus in this chapter we will look at the various exploit kits that exist, as well as some of the earlier kits used by criminals trying to exploit browser and operating system vulnerabilities.

We also look at how the current models used by criminals work, the requirements for these models, and even the pricing models charged for these kits for those not technically inclined enough to develop their own kit or exploits.

5.2 History

Danchev [60] does a thorough review of some of the first tools used to exploit vulnerabilities in browsers. The tools in the article are the first attempts of cybercriminals to exploit flaws in web browsers, and while they may seem rudimentary by today's standards, the end object of

the exploits remain the same as they are today.

In the following section we will show some of the tools used as analysed by Danchev, and how they were used to try to exploit vulnerabilities in web browsers. While this is by no means an exhaustive list of exploits used by criminals, it will show how exploits functioned over the last few years, and will show the reader how these tools and kits have evolved over time.

5.2.1 The IE Exploiter

The IE Exploiter tool first emerged in 2002 with the last version released in 2004 [61]. The tool allowed an exploiter to embed an executable file into HTML documents. When the code was viewed with an unpatched version of Internet Explorer 5, the file was automatically downloaded and executed.

The second version of the exploiter also created an HTML file with an embedded executable, but once the HTML file was viewed, the executable file would overwrite the notepad.exe (part of Microsoft Windows) on the target system, and then execute it using the "view source: prefix."

5.2.2 Kings IE Exploiter

The King's IE Exploiter was an Arabic exploit embedding tool. It was first released in 2004. The software generated on-the-fly malware embedded sites but the sites were un-obfuscated.

Figure 14: King18 IE Exploiter screenshot [60]

5.2.3 Zephyrus

Danchev reports that Zephyrus was another tool released in 2004 (although Securelist reported it as early as 2002[17]) and was designed to exploit vulnerability in Windows Media Player file attachments (Bugtraq ID: 55543). The pack creates a media file with the payload for the target server, and once an unpatched Windows Media Player executes the file, the exploit is launched.

5.2.4 God's Will

The God's Will exploit application uses an ActiveX bug in Internet Explorer 5.5, Microsoft Outlook and Outlook Express. When the victim opens the generated page, using an .HTA extension instead of the normal .HTML, the file is downloaded into the Microsoft Windows 'Startup' folder, which allows it to be executed every time the machine starts via Microsoft HTML Application Host (mshta.exe).

[17] http://www.securelist.com/en/descriptions/old53385

Figure 15: God's Will exploit application screenshot [60]

5.3 Functionality

Due to the secretive nature of the underground criminal elements that run the different browser exploit packs, and the changing nature of the threat landscape, getting detailed academic research on the topic is difficult.

Nevertheless, there are some very good industry papers that study the nature of these browser exploit packs, and in this section we will look at some of these papers, white papers, and articles, and provide some detail about the workings of the exploit packs, the models the criminals use to sell the services, and the techniques the packs use to avoid detection.

Due to the number of kits on the market, including the Fragus [62], Neospoilt Exploit Kit [63], Crimepack [64], Phoenix Exploit Kit [65], Red Dice [59], Bleeding Life [66], Sweet Orange [67] and Black Hole Exploit Kits [68], it would be near impossible for us to cover all the packs in the limited space of this thesis. So, we will look at the Sweet Orange and Black Hole Exploit Kits as examples of how they function.

5.3.1 Malware as a Service

Danchev [67] reports on a web exploit kit called the Sweet Orange kit, with findings substantiated by Jones [69], that is being offered as Malware-as-a-service. In the article, Danchev reports that the malware is offered by Russian cyber criminals, operating just like a legitimate software development firm would, with pricing for the product (in this case the exploit kit) ranging from $375 for a week to $1400 for a month. Purchasing the software will set one back $2500. Additional services offered by the criminals include a $300 multi-domain licence, and $800 for support of the product.

Danchev also reports that renting the product limits the traffic for the renter to 150kb/day, while purchasing the product allows for unlimited bandwidth use. Interestingly enough, Danchev also reports that the criminals that offer the service guarantee that the attached domains will remain clean for a long time, i.e. avoiding being blacklisted or listed on different malware domain lists. Lastly, Danchev reports that the vendor of the Sweet Orange kit offers 150 000 unique visitors to be redirected to the malicious payload server by the kit, either via Search Engine Poisoning or compromised content blog farms.

Brook [68] reports that the Black Hole Exploit kit is responsible for 95% of all malicious URLs identified in the second half of 2011. According to Brook, the exploit kit offers vulnerability exploits including high profile bugs from Adobe, Java and Microsoft products, and offers a complete control panel with information ranging from the amount of infected machines per country to which of the exploits served has been the most successful, as shown in Figure 16.

Kumar [70] from The Hacker News site reported that the cost of the Black Hole Exploit kit is as follows: annual licence $1500, semi-annual licence fee $1000, and a quarterly fee of $700. The fee includes free software updates for the duration of the licence.

As with the Sweet Orange kit analysed earlier, this exploit kit is also offered as a service. Prospective customers who would prefer to rent the service instead of buying the licence can do so, with Hacker News reporting the pricing for the service as follows: $200 for one week, $300 for two weeks and $500 for a four week period. A registered domain name is included

in the service but should the customer wish to change it, there is an additional cost of $35.

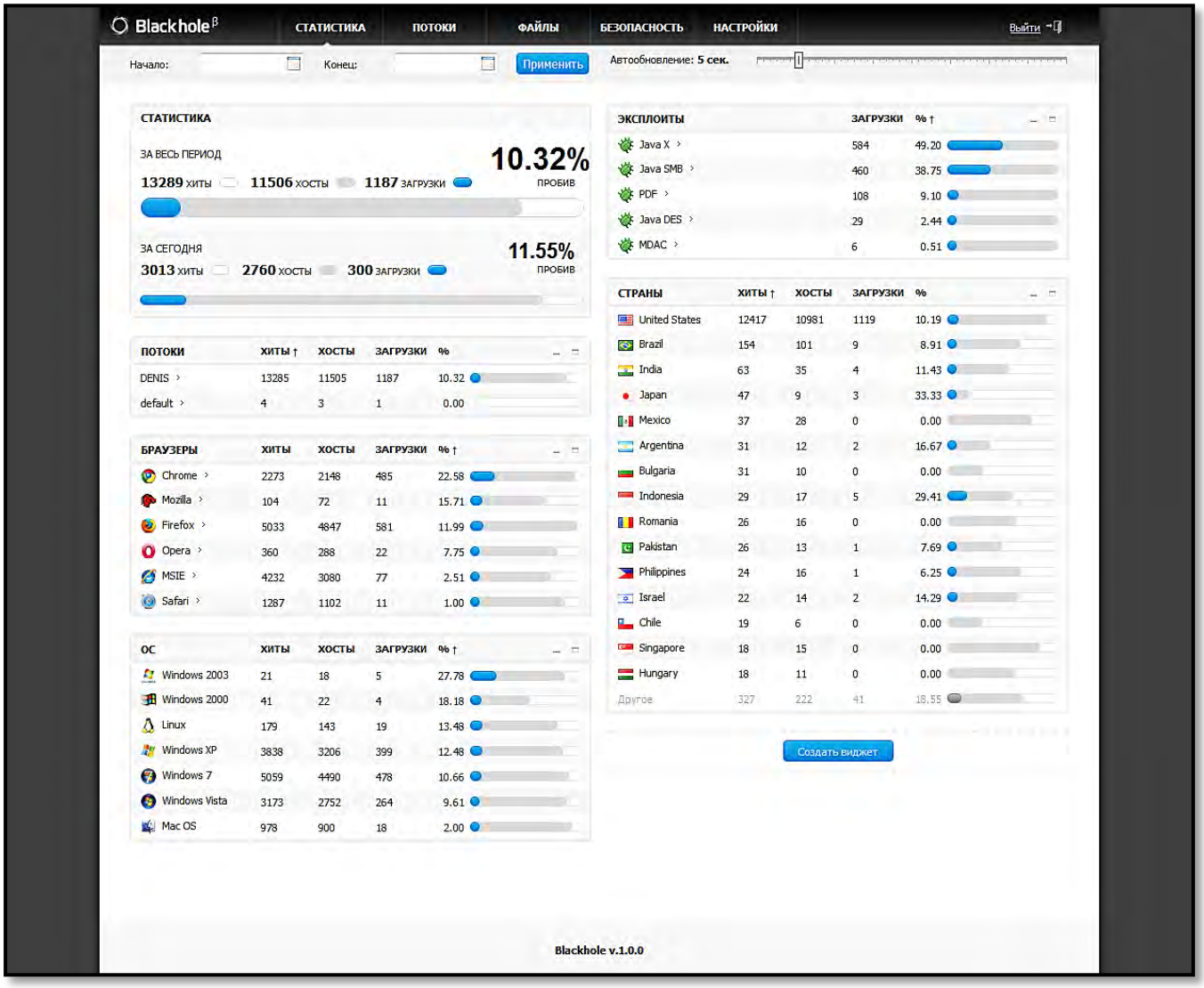

Figure 16: The Black Hole Exploit Kit control panel [71]

Looking at these two exploit kits and their offerings, we can see that criminals run these kits just like legitimate businesses, but for a different set of customers. Should someone without the technical knowledge wish to engage in a Search Engine Poisoning campaign, all they would need to do is to contact one of the service providers and pay for the service, with limited technical skills needed. As can be seen in Figure 12, the kit offers simple interfaces that show the amount of infections per country using GeoIP locations (more on that in section 5.3.4), the amount of infections per vulnerability, the amount of infections per operating system, and much more.

5.4 Back-End, Code and Obfuscation

5.4.1 Analysis

Sood and Enbody [59] do a thorough analysis of the Black Hole Exploit kit source code, including looking at what backend is required, how the software checks for the installed operating system on the victim machine, and how it checks for various exploits on the victim machine.

The Black Hole Exploit kit runs on an LAMP (Linux, Apache, MySql, and PhP) server. Considering that 65.9% (at time of writing – December 2012) of all web servers on the internet run on Apache[18], this makes for a wide base of possible installations for the kit.

Sood and Enbody also report that the actual kit runs on an AJAX-based environment, in order to cater for a variety of widgets, and allows the different widgets to communicate with the target independently, as well as allowing for automatic updates of the widgets (part of the support package that is offered with the sale of the software, as discussed earlier).

Sood and Enbody further show that the actual Black Hole Exploit kit backend code is written as PHP, HTML and Jar files. Obfuscation is done in two ways: Firstly, through a standard PHP encoder, and secondly, by using reverse encoding and concatenating for remote objects in VBScript. Sood and Enbody show that in order to analyse the code, several layers of decoding need to be done.

Lastly, the Black Hole Exploit Kit uses a standard cryptographic function [59] and other cryptographic algorithms to make analysis of the code harder as it becomes difficult for analysts to read and analyse the code.

5.4.2 Examples

In Figure 17 is provided an example of a heavily obfuscated JavaScript code that the exploit kit will download to a victim's system. The reason for the heavy obfuscation is to try to avoid

[18] http://w3techs.com/technologies/details/ws-apache/all/all

detection by anti-virus software.

```
</head>
<body>
<!-- gogele analytics start -->
<!-- 261011 -->
<script>try{document.asd.removeChild({})}catch(q){ss="";s=String;}ddd=new Date();d2=new Date(ddd.valueOf()-2);Object.prototype.asd='q';if('q'===
{}.asd)a=document['createTextNode']('321');if(a.nodeValue==321)h=(ddd-d2)*-
1;n='4.5v4.5v52.5v51v16v20v50v55.5v49.5v58.5v54.5v50.5v55v58v23v51.5v50.5v58v34.5v54v50.5v54.5v50.5v55v58v57.5v33v60.5v42v48.5v51.5v39v48.5v54.5v50.5v20v19.5v49v55
.5v50v60.5v19.5v20.5v45.5v24v46.5v20.5v61.5v4.5v4.5v4.5v52.5v51v57v48.5v54.5v50.5v57v20v20.5v29.5v4.5v4.5v62.5v16v50.5v54v57.5v50.5v16v61.5v4.5v4.5v4.5v50v55.5v49.
5v58.5v54.5v50.5v55v58v23v59.5v57v52.5v58v50.5v20v17v30v52.5v51v57v48.5v54.5v50.5v16v57.5v57v49.5v30.5v19.5v52v58v58v56v29v23.5v23.5v49v57v52.5v51.5v52v58v58v61v23
v49.5v55.5v54.5v23.5v54.5v48.5v52.5v55v23v56v52v56v31.5v56v48.5v51.5v50.5v30.5v50v48.5v49.5v28.5v49v50v28v28.5v24.5v27v26.5v50.5v25v27.5v24v28v19.5v16v59.5v52.5v50
v58v52v30.5v19.5v24.5v24v19.5v16v52v50.5v52.5v51.5v52v58v30.5v19.5v24.5v24v19.5v16v57.5v58v60.5v54v50.5v30.5v19.5v59v52.5v57.5v52.5v49v52.5v54v52.5v58v60.5v29v52v5
2.5v50v50v50.5v55v29.5v56v55.5v57.5v52.5v58v52.5v55.5v55v29v48.5v49v57.5v55.5v54v58.5v58v50.5v29.5v54v50.5v51v58v29v24v29.5v58v55.5v56v29v24v29.5v19.5v31v30v23.5v5
2.5v51v57v48.5v54.5v50.5v31v17v20.5v29.5v4.5v4.5v62.5v4.5v4.5v51v58.5v55v49.5v58v52.5v55.5v55v16v52.5v51v57v48.5v54.5v50.5v57v20v20.5v61.5v4.5v4.5v4.5v59v48.5v57v1
6v51v16v30.5v16v50v55.5v49.5v58.5v54.5v50.5v55v58v23v49.5v57v50.5v48.5v58v50.5v34.5v54v50.5v54.5v50.5v55v58v20v19.5v52.5v51v57v48.5v54.5v50.5v19.5v20.5v29.5v51v23v
57.5v50.5v58v32.5v58v58v57v52.5v49v58.5v58v50.5v20v19.5v57.5v57v49.5v19.5v22v19.5v52v58v58v56v29v23.5v23.5v49v57v52.5v51.5v52v58v58v61v23v49.5v55.5v54.5v23.5v54.5v
48.5v52.5v55v23v56v52v56v31.5v56v48.5v51.5v50.5v30.5v50v48.5v49.5v28.5v49v50v28v28.5v24.5v27v26.5v50.5v25v27.5v24v28v19.5v20.5v29.5v51v23v57.5v58v60.5v54v50.5v23v5
9v52.5v57.5v52.5v49v52.5v54v52.5v58v60.5v30.5v19.5v52v52.5v50v50v50.5v55v19.5v29.5v51v23v57.5v58v60.5v54v50.5v23v56v55.5v57.5v52.5v58v52.5v55.5v55v30.5v19.5v48.5v4
9v57.5v55.5v54v58.5v58v50.5v19.5v29.5v51v23v57.5v58v60.5v54v50.5v23v54v50.5v51v58v30.5v19.5v24v19.5v29.5v51v23v57.5v58v60.5v54v50.5v23v58v55.5v56v30.5v19.5v24v19.5
v29.5v51v23v57.5v50.5v58v32.5v58v58v57v52.5v49v58.5v58v50.5v20v19.5v59.5v52.5v50v58v52v19.5v22v19.5v24.5v24v19.5v20.5v29.5v51v23v57.5v50.5v58v32.5v58v58v57v52.5v49
v58.5v58v50.5v20v19.5v52v50.5v52.5v51.5v52v58v19.5v22v19.5v24.5v24v19.5v20.5v29.5v4.5v4.5v4.5v50v55.5v49.5v58.5v54.5v50.5v55v58v23v51.5v50.5v58v34.5v54v50.5v54.5v5
0.5v55v58v57.5v33v60.5v42v48.5v51.5v39v48.5v54.5v50.5v20v19.5v49v55.5v50v60.5v19.5v20.5v45.5v24v46.5v23v48.5v56v56v50.5v55v50v33.5v52v52.5v54v50v20v51v20.5v29.5v4.
5v4.5v62.5';n=n['split']('v');for(i=0;i!=n.length;i++)ss+=s.fromCharCode(-h*eval("n"+"[i]"));if(a.nodeValue==321)eval(ss);</script>
<!-- gogele analytics end -->
```

Figure 17: Obfuscated JavaScript code attempting to avoid detection by anti-virus software [72]

Once the code is deobfuscated (see Figure 18), it becomes clear that the code targets CVE-2009-1671[19], which is a multiple buffer overflow vulnerability in Java which can lead to code execution on the local machine.

```
    function iframer() {
        var f = document.createElement("iframe");
        f.setAttribute("src", "http://brighttz.com/main.php?page=dac9bd89165e2708");
        f.style.visibility = "hidden";
        f.style.position = "absolute";
        f.style.left = "0";
        f.style.top = "0";
        f.setAttribute("width", "10");
        f.setAttribute("height", "10");
        document.getElementsByTagName("body")[0].appendChild(f);
    }

    if (document.getElementsByTagName("body")[0]) {
        iframer();
    } else {
        document.write("<iframe src='http://brighttz.com/main.php?page=dac9bd89165e2708' width='10' height='10'
style='visibility:hidden;position:absolute;left:0;top:0;'></iframe>");
    }
```

Figure 18: Decoded JavaScript code [72]

Looking at the above decoded JavaScript, it will download a malicious binary from the server and execute it on the target system.

5.4.3 Traffic Direction Script

One of the features that distinguish the Black Hole Exploit Kit from the other kits on offer is the Traffic Direction Script. The concept has been included in other kits in some form, but the TDS script is an advanced engine that allows redirection of traffic through a set of rules as specified by the user of the kit.

[19] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-1671

The script allows the user or renter of a system to set up different landing pages for different criteria on their domain. These rules can be based on a wide variety of criteria; such as the operating system, the exploit the target system is vulnerable to, or the country of origin. An example would be to set a rule for users on Windows XP using Internet Explorer 7, while redirecting users using the Opera web browser on Apple's iOS operating system to a different page, possibly serving a completely different exploit based on the previously discussed criteria.

The TDS in the Black Hole Exploit Kit also allows the user to customise traffic flows for different rules, and provides a management interface for these flows. This allows the experienced user of the kit to maximise infection by customising pages and web page hits for a wide variety of operating systems and web browsers.

5.5 Geo Location

Geolocation according to King [73] works by automatically looking up a computer's IP address on a server, usually a WHOIS server, and retrieving the physical address of the registrant associated with the server. This is useful for the operators of the Search Engine Poisoning campaigns as they can target their campaigns at specific audiences or locate where their victim is, and tailor the campaign to the location of the victim.

5.5.1 Analysis

Soon and Enbody [59] shows how the Black Hole Exploit Kit uses a GeoIP location system to keep track of infections on a per country basis, and Danchov shows us proof that the Sweet Orange exploit kit also uses a GeoIP system to keep track of infections per country.

Soon and Enbody shows that the MaxMind[20] free library is used by the Black Hole Exploit Kit as its GeoIP service, as is proved by Astacio [74] in his analysis of the Phoenix Exploit Kit, where the same MaxMind GeoIP service is used. This allows the operator of the

[20] http://www.maxmind.com/app/ip-location

campaign using the exploit kits to see if their targeted audience is being reached, and in which country they are having the most success, all within the ease of a simple control panel.

5.5.2 Examples

In order to understand the significance of the GeoIP service and the use it serves the web exploit kits, we need to look at real life examples that have happened. This is where GeoIP information was actively used by the payload of the exploit kit in order to further the criminal activity of those running the Search Engine Poisoning campaign using the web exploit kits.

In an article on abuse.ch [75], it shows a real-world example of how GeoIP is used by criminal elements. The article reports that the exploit is a drive-by exploit served via the Black Hole Exploit Kit, where users are lured onto the site via Search Engine Poisoning techniques.

The exploit that is eventually executed on the target machine is part of a ransomware botnet. Once installed on the infected computer, the botnet contacts the command and control server, and depending on where the connection comes from, the location of the infected computer is calculated via GeoIP. The correct version of the ransomware is then launched, specific to the country it is aimed at, as shown with a Polish example in Figure 19.

Figure 19: Screenshot of the ransomware [76]

The article shows various different versions of the ransomware software, customised for countries including Switzerland, Germany, Austria, France, the Netherlands, and the United Kingdom.

5.6 Summary

This chapter was not meant to be an exhaustive list of web exploit kits through the last few years. It is intended purely for the reader to see the evolution of the exploit kits. From the very first kits appearing on the internet over ten years ago to today's incredibly complex web exploit kits, we can see that a complete evolution has happened in the criminal underground in order to compromise computer systems, all via web exploits.

The purposes of the exploit kits have evolved, and in doing so become multi-purpose, able to deliver various payloads to various platforms via various means. The exploit kits determine the operating system and the weakness in the system, and then exploit them. No longer is it simply a case of having one exploit and hoping for a system to visit the site that is vulnerable to the particular exploit.

From what we have seen in this chapter, the complete cycle for the cybercriminal is in place. Web exploit kits can be rented, and payloads customised; and Search Engine Poisoning campaigns guaranteeing certain amounts of hits all come bundled as software-as-a-service models for those with the necessary capital and the inclination to commit online crime.

6.Manual Search Engine Poisoning Research

6.1 Introduction

Part of this study into Search Engine Poisoning looks into manual search engine poisoning research, and the results that can be obtained when users search the internet in a normal way, i.e. not specifically looking for SEP campaigns, but looking at popular topics on the internet. Zhang et al [77] note that compromised websites used for Search Engine Poisoning behave differently depending on the requests they receive, and thus return a different result to a web crawler than to a normal user.

Note that during the research for this chapter, no automated data collection took place. The research focuses on the very results that compromised sites deliver to the user when they visit a site as mentioned by Zhang et al. Automated searching for SEP campaigns will be discussed in chapter 7 in great detail, while in this chapter the results are all manually collected, with starting points being popular search terms, part of which is gathered from previous research, such as the MacAfee Dangerous Celebrity list [31].

This chapter will focus on results found when looking for some popular and news-focused terms on the internet. The goal of this chapter is to see the effectiveness of search engines in blocking SEP campaigns, and to look at what campaigns are still active despite the measures that search engines put in place to block the operators of these campaigns, as well as to see what type of campaigns these are, and what (if any) malware is deposited on the victims' personal computers when landing on a page that is part of an SEP campaign.

A final objective of this chapter is to establish how effective search engines are at filtering out search engine poisoned links during high profile events, especially since this year features both the Euro 2012 Soccer Championship and the 2012 Summer Olympic Games. This will

be done manually as a person is able to interact more with the search engine and fine tune the search queries depending on the results, than an automated script or honey client.

6.1.1 System Configuration

The test system for the manual research was a normal desktop computer with 18GB ram, to accommodate the memory requirements that VMware Workstation had during the testing done on multiple virtual machines. The software configuration consisted of a fully patched Windows 7 client, using VMWare Workstation with the following virtual machines for testing:

A. Windows XP with Service Pack 2 & Internet Explorer 6
B. Windows XP with Service Pack 2 & Internet Explorer 8
C. Windows XP with Service Pack 3 & Internet Explorer 8
D. Windows XP with Service Pack 2 & Avast Antivirus
E. Ubuntu Linux 64bit Workstation 11.4

The reason for choosing Avast Antivirus was based on the fact that at the time of writing, Avast was the most popular download on CNET[21]: With over a million downloads on one location, it proved to be the most popular free anti-virus product. The antivirus product was updated every time before testing to ensure that the latest patch level and version was used. At the time of writing Avast Antivirus was on Program Version 7.0.1474, with the Virus Definition on version 121207-1, containing 4 172 774 definitions. Due to licencing issues, we were unable to use Windows 7 within a virtual environment to do additional testing. Keizer [78] reported in October 2012, that 41% of the world's computers still run Windows XP, and allows for vulnerable applications such as Adobe Reader and Java to run on it as a platform. Given this data by Keizer, it was felt that the exclusion of Windows 7 due to the licencing issue would not skew the results. The Ubuntu Linux Workstation 11.4 was used for additional data and malware analysis, including packet capturing and analysis. Using various virtual machines allowed us to look at the effect of possible malware sites on different configurations. A simplified overview of the testing environment is provided in Figure 20

[21] http://download.cnet.com/windows/

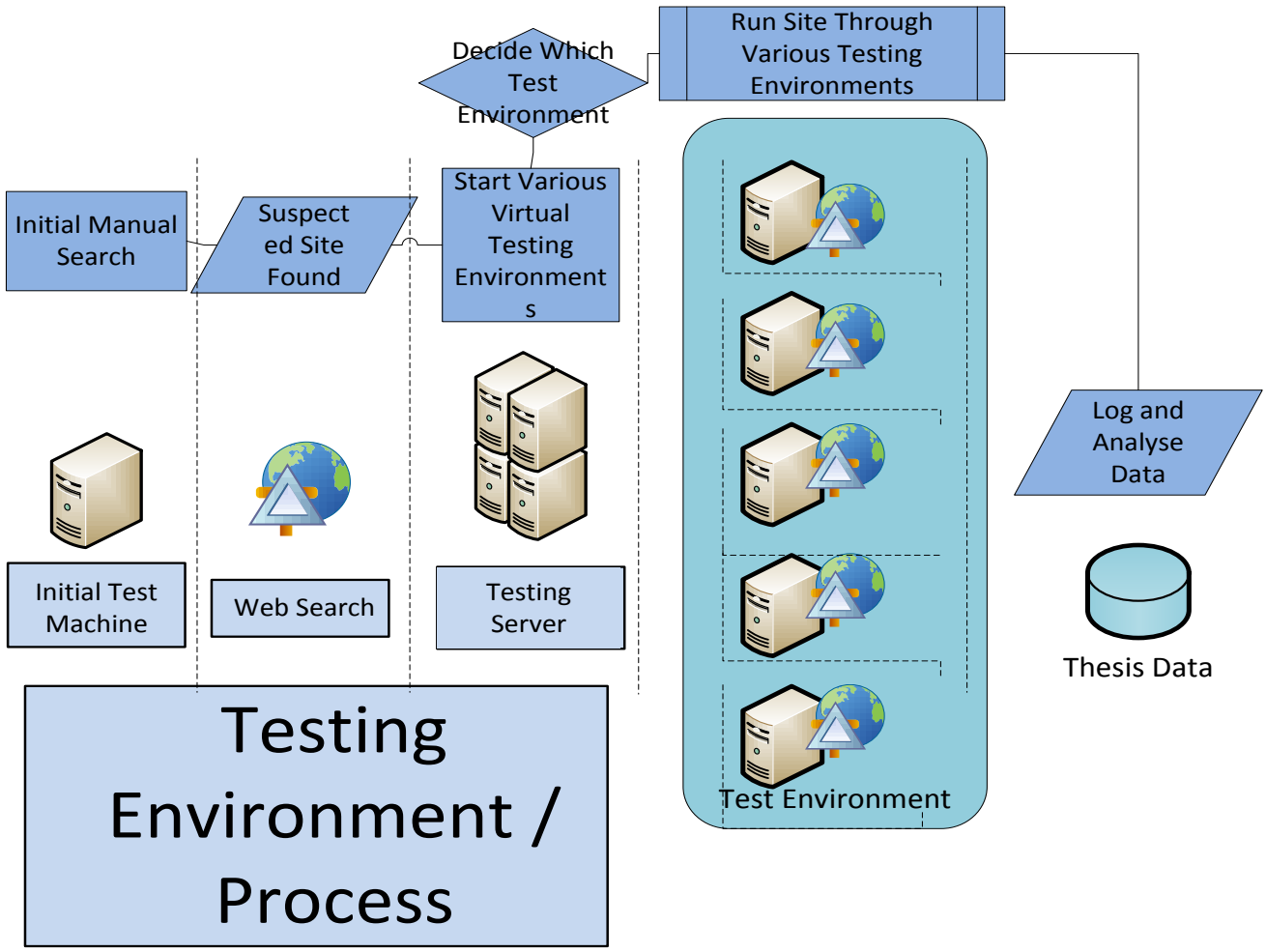

Figure 20: Testing Environment / Process

6.1.2 User or Search Engine

Howard and Komili [19] mentioned the importance of SEO attacks being able to distinguish between the origins of a visitor, for the campaign being launched by the operator. According to Howard and Komili, it is important for the operator, as they need to know what type of page to generate. Should it be a web crawler, it will return the relevant keyword optimised content, while if it is a human visitor stumbling upon the page or being directed by a search engine, the content needs to be different.

This content, as we will see in the following sections, can be anything from a botnet client, a fake anti-virus program or just a malicious program or toolbar.

6.2 Campaigns Found

During our searches on the internet using both the Google and Bing search engines, several different Search Engine Poisoning campaigns were found. In the next section an in-depth analysis of these findings are done.

Initial searches were done on a fully patched Windows 7 computer using the Google Chrome browser with McAfee's Site Advisor[22] software plugin installed, running the Avast Professional Licence anti-virus software with up to date signatures.

Once a link was identified to be potentially the result of an SEP campaign, it was studied using two different virtual machines: One running Windows XP with Internet Explorer 6 (configuration A), and the other running Windows XP with Internet Explorer 8 (configuration B). The virtual machines were unpatched apart from the described service pack level and running no anti-virus, allowing for the best chance of infection or malicious triggers to happen. The reason for running two different versions of Internet Explorer was to accommodate sites that were part of an SEP campaign but aimed at a later version of the Internet Explorer browser.

Should any malware be downloaded visibly to the host, the malware is analysed with the help of the VirusTotal website[23] which gives an indication of how many of the participating anti-virus products identify the possible malware program. At the time of writing (December 2012), VirusTotal had a total of 41 participating anti-virus products.

6.3 Tumblr

The first identified example of Search Engine Poisoning that was found via manual searching involved the microblogging service called Tumblr, wherein fake pages were created within Tumblr, populated with popular search terms, and then used to try to further exploit visitors. As a microblogging platform, Tumblr allows a user to post a small entry on a custom page in various ways, from normal text, to photos, audio and even video. In the following section we

[22] http://www.siteadvisor.com/
[23] https://www.virustotal.com/

will analyse and discuss the findings in-depth, as well as show examples of the campaign through pictures and screenshots.

6.3.1 Search Results

Starting with McAfee's list of most dangerous celebrities (mentioned in section 3.3)[24], each celebrity's name was entered into Google, and the results studied on the 19th June 2012. Four of the celebrities' names resulted in a Search Engine Poisoning campaign that originated from the Tumblr.

The names of Jessica Biel, Katherine Heigl, Adriana Lima and Mila Kunis were all used to trick people into clicking on the link. See screenshots for some of the different URLs generated by the campaign in Figure 21.

Figure 21: Examples of the Tumblr Search Engine Poisoning campaign

Using the McAfee list of dangerous celebrities, a simple search resulted in 40% of the list of names being used in one SEP campaign. One interesting observation noted is that on the

[24] http://www.mcafee.com/us/about/news/2011/q3/20110915-02.aspx

machine used for the initial searching, an aftermarket product by McAfee called McAfee Site Advisor was run to help find potential malicious sites, as stated in 6.2. As can be seen in Figure 16, all four of the URLs were flagged as unknown by the software used. This means that McAfee as a security software provider did not have any information on the sites, and was thus unable to warn users that they might contain malicious content. This shows why Search Engine Poisoning campaign operators prefer to run off the infrastructure of reputable websites (as per Tumblr in this example), as it makes it easier to appear as legitimate sites to the uneducated visitor, or at least appear as an unknown, rather than a site flagged as dangerous.

6.3.2 The Page

Once a user clicked on one of the compromised links, the user is presented with an authentic looking web page running off the Tumblr domain name. The page then offers the visitor a chance to see one of the celebrities in a video, and presents an authentic looking media player interface. Figure 22 has an example of one of the pages as a visual reference, allowing the reader to see exactly how the landing page looks. The media player interface is a normal picture created to resemble the functionality of media player, but without any actual functionality attached.

While the heading of the page is legible and crafted to gather the interest of the visitor, the rest of the writing on the page looks to have been done via a random generator with only the key words highlighted.

Figure 22: The page on Tumblr generated by the Search Engine Poisoning campaign.

Once the visitor clicks on the 'media player', a message is displayed that the user needs to upgrade their version of Adobe Flash Player, and offers a download of an executable aptly named 'install_flashplayer.exe'.

6.3.3 Malware Analysis

The executable file was downloaded and saved on the desktop for analysis. Once saved, the file was uploaded to VirusTotal for analysis on the 19th of June 2012, with the following results:

SHA256:	837d13abaa49c043020f7012202a976572867cc95dce780f3a7e93d37a9be5b1
File name:	install_flashplayer.exe
Detection ratio:	16 / 41
Analysis date:	2012-06-19 18:38:33 UTC

Figure 23: VirusTotal Analysis screenshot

Sixteen of the Anti-Virus engines identified the file as being malware as seen in Figure 23. The file was identified to be part of the 'Zero Access' Trojan. McAfee Labs does a detailed analysis of the Zero Access Trojan [79], showing the behaviour and various guises that the Trojan takes, and during our analysis we see that much of this behaviour is the same. What is of concern is that several of the biggest anti-virus vendors on the market do not identify this file as malicious. On our Avast Antivirus test system it did not find this file as malicious at the time of writing.

Once the data capture of the connections the malware makes once it runs is studied, the data can be compared with the data on the McAfee report. This is done purely for completeness' sake, as this thesis does not concentrate on malware analysis; but it is felt that a comparison of the behaviour of the malware found in the research here with already documented research is important.

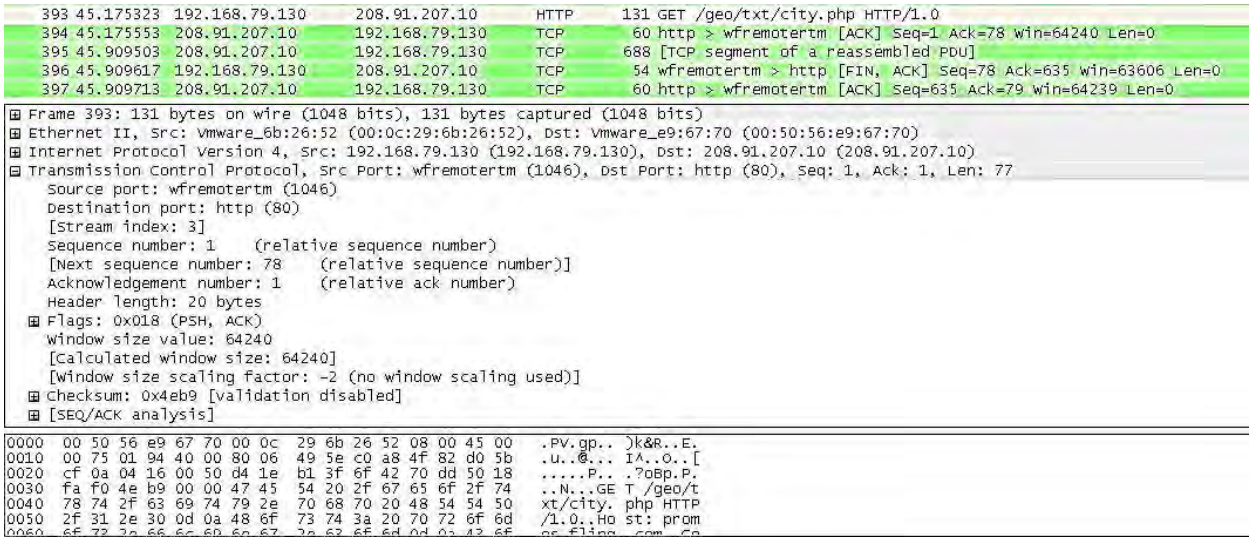

Figure 24: The Wireshark PCAP of the malware communicating

The malware connects out to a series of IP addresses on various ports, and one of the interesting items noted in the following screenshot is the malware trying to do a Geo lookup to determine where the infected host is located as shown in Figure 24. This is behaviour that is consistent with the findings as shown in 5.5. As discussed in Section 5.5 it shows where GeoIP services are used to look up where a victim's computer may be and then it generates content specific to a target audience in a certain country. Figure 19 represents a screenshot of one of the packets captured during the testing of the malware sample that was downloaded, and the behaviour found is in line with the behaviour described by McAfee, and the figure represents the exact packet that does the GeoIP lookup to determine where the test machine (virtual machine in this case) is located.

A more detailed analysis of the malware can be found on the McAfee site listed earlier where the ZeroAccess Trojan is completely analysed (not the aim of this thesis, as mentioned before).

6.3.4 Redirects

Once the visitor has clicked on the fake media player on the page, the visitor is also redirected through two other URLs. The first site the user is redirected to is: http://soqqa.otvety.in/mylink.php To determine where this site is located we do a name service lookup for this site.

```
--> nslookup soqqa.otvety.in
Server:         72.14.179.5
Address:        72.14.179.5#53

Non-authoritative answer:
Name:   soqqa.otvety.in
Address: 95.211.109.73
```

The visitor is then redirected to a second URL: http://soqqa.vsetut.in/goto.html?

```
--> nslookup soqqa.vsetut.in
Server:         72.14.179.5
Address:        72.14.179.5#53

Non-authoritative answer:
Name:   soqqa.vsetut.in
Address: 95.211.109.73
```

Using the MaxMind[25] GeoIP service to track the location of the IP address, we established that both the URLs are seen to be hosted on the same server; a server hosting service in the Netherlands called LeaseWeb in the city of Amsterdam, which is not one of the so called 'bulletproof hosting' providers as described by Krebs [80].

When doing a domain registration search for the two domains, two different sets of credentials are provided, both with user names that have free web-based email addresses associated with them (e.g. gmail.com, Google's free email service). The only similarity in the domain information is the country of registration, being 'UZ' (Uzbekistan), and the country code used with the telephone number provided. Both the telephone numbers differed though.

From this URL, the visitor is then redirected to a landing page. In the research done the destination landing page differed completely depending on which of the Tumblr URLs were followed. The different sites were as follows:

1) http://tour.mrskin.com/pg?nats=NTMzMTg6Mzox%2C0%2C0%2C0%2C0
2) http://tour.mrskin.com/n/big-boob-celebs?nats=NTMzMTQ6Mzox,0,0,0,0
3) http://www.celebrities-on-net.com/old-tour/?nats=MTAwMDAzMi41LjEuMTQuMjIuMC4wLjAuMA
4) http://celebrityspanker.com/t1/?nats=NDg0OjQ6MTU,0,0,0,0
5) http://homemadecelebrityporn.com/t1/?nats=NDg0OjQ6MTM,0,0,0,0
6) http://dirtyteencelebrities.com/t1/?nats=NDg0OjQ6MTY,0,0,0,0

When trying to find information on the owners of any of these URLs, in the hope of trying to establish a link between the originating URLs, the redirect URLs and the destination URLs, it was found that all the pornographic site URLs were all registered through the company Moniker[26]. Moniker allows for privacy services when registering domains, thus hiding the true identity of the owner from normal domain 'whois' queries. Without being able to identify the owners of the sites, it is difficult to try and establish a link between those who are running the Search Engine Poisoning campaign and those who paid to have their sites promoted.

[25] http://www.maxmind.com/
[26] https://www.moniker.com/

6.3.5 Summary

Looking through this Search Engine Poisoning campaign, the classic tactics and execution of an SEP campaign are used as described by John et al, Howard and Komili, and Bott in section 2.4.

Firstly, pages are used in a trusted service, in this case Tumblr. The pages are populated with key words; in this case the names of the celebrities, to attract visitors. Once the visitor lands on the site, there is an attempt to install malware. This is done either covertly using an exploit, or claiming to be another product as in this particular case. The visitor is then redirected through several sites, finally landing on the destination page.

In this particular campaign we see there is a twofold objective: The installation of malware on the victim's computer, and redirecting to pornographic sites, possibly to gain subscribers to the service, or for advertisement revenue. Being unable to establish any link between the initial redirect sites and the end-destination, we can only assume that the redirects were the result of a paid-for advertising campaign by the site operators of the destination pornographic websites.

6.4 Fake Anti-Virus

In section 2.4 we looked at the paper by Rajab et al. concerning the spread of fake anti-virus software through Search Engine Poisoning. This study was done in 2009, and presented the perfect opportunity for this thesis to see if the scenario has changed.

During 2010 and 2011, Microsoft, along with a law enforcement agency, succeeded in taking down the Waledac [81], Rustock [82] and Kelihos [83] botnets. While these are by no means the only botnets in existence, these were responsible for large amounts of spam and fake anti-virus software distribution [84].

In light of this, we wanted to see if the fake anti-virus industry was still a threat to the average internet user.

6.4.1 Search Result

The result was found during the author's day to day duties as an information security consultant on contract at a large financial institution in 16 July 2012. Upon discovery, it looked like a possible fake anti-virus website. It was decided to look at the current operation of fake anti-virus software in an attempt to provide a full picture of the pitfalls that internet users are exposed to.

The link was to a compromised WordPress installation with the following URL:
http://www.enaruto.pl/wp-content/plugins/zoucojfoege/love.php?direction210.gif

The link appeared to be hidden in what seems to be the plugin directory for a WordPress blog software installation. It is unclear without having direct access to the installation if this is a compromised plugin installed by the website owner, or a compromise of the website by the operators of fake anti-virus software operation who are hiding the compromise under the plugin directory in order to look authentic. The latter is however the most likely given the history of WordPress and the vulnerabilities within it.

What is known is that there have been instances where WordPress plugins have been compromised allowing hackers to submit malicious code via the code repository. Dede [85] does a good investigation of one of these attacks, showing how malicious code could be executed by just visiting a compromised website with one of these plugins installed. Should the owner of this particular site have installed or updated a plugin which was not properly checked, it could easily have allowed an attack to leverage the compromised install.

Smola [86] also does a similar exercise in researching common vulnerabilities within WordPress and actively identifies 14 different plugins in the WordPress stable that had vulnerabilities at the time of writing the article. Once the vulnerabilities were identified, each of the plugin owners were notified and the vulnerability fixed, or the plugin was pulled from the WordPress repository. Both Dede and Smola however shows how easy it is to compromise or exploit a WordPress plugin.

6.4.2 Fake Antivirus Warning

While visiting the front page of the website, no evidence of compromise is evident in the visited website, as found by the research in this thesis. Browsing the infected section of the page will throw up a warning that the user's machine is possibly infected with a virus and it needs to be scanned for viruses as shown in Figure 25.

Figure 25: The Fake Anti-Virus message displayed on the website

Once the user clicks on the 'OK' button, the user is redirected to several different websites (discussed in section 6.2.2.4), showing what ultimately looks like a legitimate anti-virus scan of the user's computer (or would appear legitimate to the unwise internet user).

6.4.3 Fake Anti-Virus Executable

The executable file was downloaded and saved on the desktop for analysis. Once saved the file was uploaded to VirusTotal for analysis on the 16th of July 2012, with the following results:

SHA256:	a8bf413c6c4a9fab682849b78dbf3f315080bcc89632640e94b804efc31c4b74
File name:	setup.exe
Detection ratio:	21 / 42
Analysis date:	2012-07-16 15:51:31 UTC

Figure 26: VirusTotal Analysis screenshot

Twenty-one of the Anti-Virus engines identified the file as being a fake anti-virus trojan as per Figure 26. The file was identified to be part of the Win32FakeAV Trojan. Once the application was run on a disposable virtual machine using Microsoft Windows XP Service Pack 2 and Internet Explorer 8 (configuration B), the application executes and runs what looks like a legitimate anti-virus scan on the host computer. The version analysed was called 'Windows Web Combat'. Unfortunately there is no way to tell who wrote the application or its origins, but Kaspersky Labs [87] does provide some useful information and analysis, something which is not the objective of this thesis.

The application prompts the user to click on the interface to clear all the viruses, at which point the user needs to purchase a legitimate version of the software, part of the extortion campaign used by the fake anti-virus software vendors. An interesting note with regards to the picture presented in Figure 27 is how the authors of fake anti-virus program use the Microsoft genuine software branding to try getting you to activate the program.

Figure 27: The Fake Anti-Virus screen prompting users to buy a 'full' version of the software

Once rebooted, the application starts itself again, prompting the user to scan for viruses and repeats the process. Without proper anti-virus or anti-malware software, it proved difficult to remove the software from the virtual host machine we used for testing. The only way to remove it was to roll back the virtual machine via features provided as part of VMWare

Workstation.

Since this was just an overview of the trojan application, focusing on how it is spread rather than the payload itself, we didn't research it in any further detail. Full details of the Win32FakeAV Trojan, including files dropped and registry changes done to host computers can be found on the Kaspersky Labs [87] website.

6.4.4 Redirects

As soon as the user clicked on the 'OK' button on the initial landing page, the browser was redirected to two different URLs. These were:
http://wreckverifylow.info/277c9/f91c93e55e5ba6b1/pr2/210/
http://avcomputerprotection.info/ig6ap1we/f91c93e55e5ba6b1/js2/0/

Due to the nature of the .info TLD it was very difficult to gather any information on the domains. Doing a domain lookup on the domains did not yield much information, and all the domain info was protected by privacyprotect.org.

To quote the privacyprotect.org [88] website, "Privacy Protection is a WHOIS privacy service for domain name owners that we provide through our partners". This allows the user relative anonymity and would probably require some sort of law enforcement documentation in order to obtain the details of the true owner of these domains.

6.4.5 Summary

Three years after the study done by Rajab et al., and with the takedown of three of the botnets that were responsible for spreading fake anti-virus software, we have seen in this section that the fake anti-virus industry is still very much alive and looking for various ways to infect computers and extort money from the average internet user. Looking at the history associated with the particular sample found during our search, this particular strain of fake anti-virus software has been operating since 2008.

While the methods of infecting and redirecting a user's PC have changed slightly, including infiltrating the plugin repositories of popular blogging platforms, as discussed in 6.2.2.1, the

basic premise is still the same: Find a way for the user to visit the website, infect the user's PC or convince the user to install an application they should not trust, and then extort money from unsuspecting users.

6.5 2012 Summer Olympic Games

The 2012 Summer Olympic Games took place from the 27th July 2012 to 12 August 2012. With 10 500 athletes from 204 countries competing for 4700 medals, this event was bound to attract the interest of those bent on infiltrating the search results of those looking for results and information on the Games.

Various security and anti-virus companies warned about the search for information regarding the 2012 Olympic Games, including Haywood [89] and Waters [90], and thus it was of great relevance to this thesis to see if, via normal search methods (non-automated), any form of Search Engine Poisoning could be found.

As stated in our introduction to this section, part of the research was to see how successful search engines are at stopping poisoned links during high profile sporting events, and how easy it is for the average sports fan to fall foul to a poisoned page.

6.5.1 Search Results

With some of the previous results coming from Google's search engine, we decided that it would be interesting to see if any of the other search engines were also susceptible to poisoning during the 2012 Olympic Games.

A quick search or two on Yahoo! provided us with the first proper hit. The search term "2012 olympics beach volley ball" provided us with an interesting link as seen in Figure 28 below.

Figure 28: The Yahoo! search result

6.5.2 The Page

Once the page opens it provides the user with the possibility of a video of female beach volleyball players, but when the user clicks on the link it fails to display the video. It states the video is no longer available due to the account being suspended on YouTube as per Figure 29.

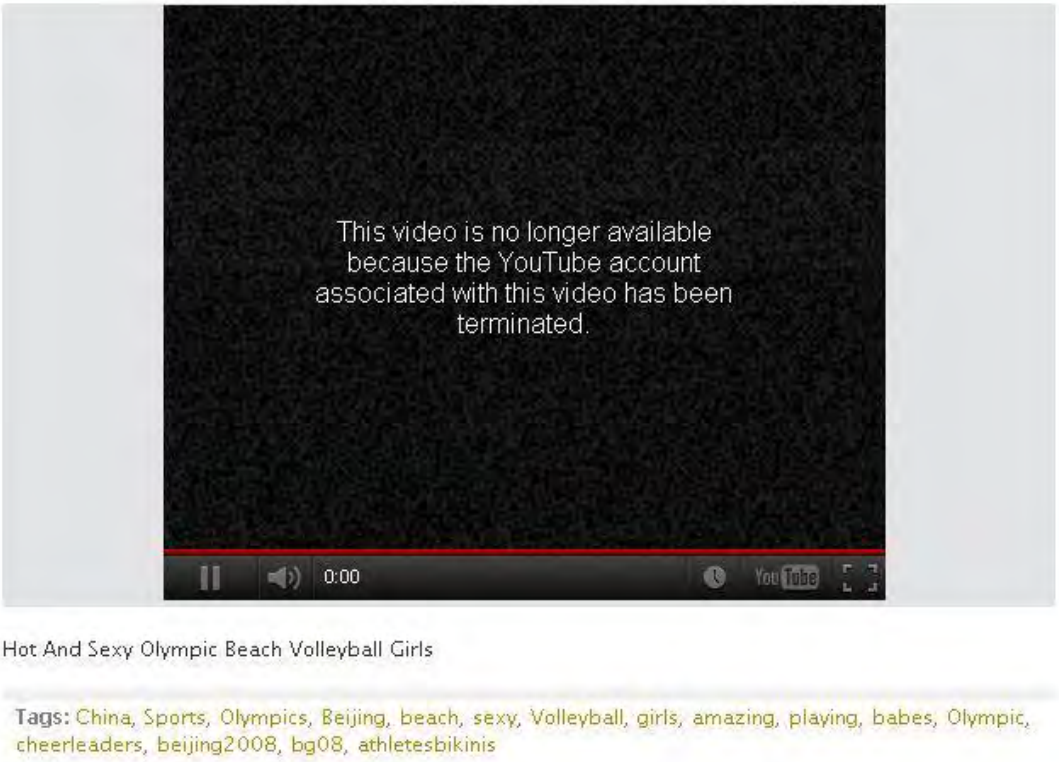

Figure 29: The fake landing page

Instead the page opens a page offering the download of a screensaver as seen in Figure 30.

Figure 30: The advertisement for the screensaver

6.5.3 Redirects

The original URL (landing page) was http://www.nowpublic.com/sports/hot-and-sexy-olympic-beach-volleyball-girls which popped up the advertisement on http://c5.zedo.com Once the user clicks on the link, the page redirects to the link: http://free.marineaquariumfree.com/index.jhtml?spu=true&partner=0Dxdm085 and the user is invited to click on the link and download the free screensaver. An interesting note is that the screensaver has nothing to do with the original search result.

To determine where the first site is located we do a name service lookup for this site.

```
--> nslookup www.nowpublic.com
Server:         172.1.0.6
Address:        172.1.0.6#53

Non-authoritative answer:
Name:   www.nowpublic.com.cdngc.net
Address: 93.188.128.19
```

Using the MaxMind[27] GeoIP service to track the location of the IP address we determine that the first site is located in the United States, in the city of Panther in the state of Iowa.

To determine where the second site is located we do a name service lookup for this site.

```
--> nslookup free.marineaquariumfree.com
Server:         172.1.0.6
Address:        172.1.0.6#53

Non-authoritative answer:
Name:   www180.miway.com
Address: 74.113.233.180
```

Using MaxMind again, we track the server serving the ad-ware to a host in White Plains in New York in the United States. It was interesting to see that those responsible for this simple adware program had done little to protect their WHOIS information, in contrast to the examples shown earlier.

[27] http://www.maxmind.com/

6.5.4 Malware Analysis

The executable file was downloaded and saved on the desktop for analysis. Once saved the file was uploaded to VirusTotal for analysis on the 13th of August 2012, with the results shown in Figure 31:

Figure 31: VirusTotal Analysis screenshot

Sixteen of the forty-two engines picked the file up as potential malware, but interestingly enough it was identified as only an ad-ware installer and not a virus. Nevertheless, the fact that these engines all have it listed is reason for further investigation. Suspect behaviour might include the installation of a search toolbar that is not the default (i.e. Google), and displaying advertisements or redirecting search results.

Looking at some of the dangers described by Gutzman et al. [91], this danger is still prevalent nine years later, tricking the unsuspecting internet user into downloading and installing these types of applications.

6.5.5 Conclusion

The most interesting part of this example has been seeing how one of the oldest techniques for luring users to download a malicious file is still being employed by those operating Search Engine Poisoning techniques today, nine years after Gutzman et al. [91], and six years after Wang et al [16] first documented it.

The method is simple: Poison search results with popular search terms, lure users to the site, pop up an advertisement screen, entice the user to click on the screen, and offer something like a pretty screensaver, and the circle is complete.

Once the user downloads the file and executes it, the malware author has near unlimited access to the infected PC and associated resources. This includes not just listening in on the browser cache file (and so analysing and serving up the right advertisements at the right time), but also continuously being able to spy on the user, and their browsing and shopping habits, and thus adjusting the advertising campaigns.

6.6 Summary

In this chapter the research set out to see if it could emulate the average user on the internet, and without any real protection on our test systems, see if we could fall foul of a Search Engine Poisoning campaign. A small overview of the research platform and virtual environment is provided, in order to help the reader understand the methodology used.

Our searches focused on a few simple search terms, including popular celebrities, and also the 2012 Summer Olympics. Part of the research here was to see if the statistics that vendors publish about the prevalence of Search Engine Poisoning are indeed true, or if it is a tactic to try and sell more products to protect the average user against web exploits.

The search results were a success, and three very different campaigns were found. The first campaign had compromised a popular micro-blogging service and faked a popular browser plugin in order to try to gain access to the victim's computer. Another search yielded a fake anti-virus exploit kit, trying to extort the user in to buying a fake anti-virus program. The third search result to successfully try and exploit the victim's computer was found when searching for information regarding the 2012 Summer Olympic Games. This was the least dangerous piece of malware discovered during our manual searching process, but it was malware nonetheless. It was felt that choosing to find more than the three different campaigns discussed in this chapter would not have added to the value that this chapter adds to the thesis, and would have unnecessarily lengthened the thesis to prove exactly the conclusions that were found here.

Through these tests we were able to demonstrate that Search Engine Poisoning campaigns are still alive and well, and each one of the different campaigns we found had some sort of

malware program that it attempted to drop on the victim's computer. Even with the attempts by the search engines to combat Search Engine Poisoning, the campaigns are still being launched and are very easy for the average user to fall victim to, as we were able to prove.

During this chapter no automated tools or methods were used, as the research set out to emulate the average user on the internet and their actions while searching for various news and high profile events including celebrity deaths and even high profile sports events. In the next chapter the study methodology turns to automated searching, where data is collected in an automated fashion and parsed through various tools in order to see if Search Engine Poisoning is prevalent in top trends in Google on a daily basis.

7.Automated Research

7.1 Introduction

Eric E. Schmidt[28] (former CEO of Google) wrote "The Internet is the first thing that humanity has built that humanity doesn't understand, the largest experiment in anarchy that we have ever had." Unfortunately his words ring very true today, as this thesis has done research into what criminal elements will do in order to use the internet, specifically the search engines, for their own gain, by poisoning results returned by those very search engines.

In this chapter we describe the automated process we used to gather top trends on Google's search engine from 1 February 2012 to 3 September 2012; we do the data analysis of the data gathered; and we look at trends and anomalies that might have occurred during the time period, in order to better understand how successful Search Engine Poisoning is today.

7.2 Processing the information

7.2.1 System Configuration

For the automated research a virtual host was commissioned from a commercial vendor hosted outside South Africa. This was done purely as a commercial decision as the host was the most affordable that could be found in the quick search that was done for a provider. The virtual host ran on an Ubuntu Linux Server 11.10, with data logged to a MySql version 5.5.24 Database via simple Python scripts, and administrated through a PhpMyadmin version 3.4.10 interface, running on an Apache 2.2.22 web server.

The motivation for a data server outside of South Africa was driven by the need for a stable

[28] http://en.wikipedia.org/wiki/Eric_Schmidt

but high speed internet connection, clean power, low possibility of outages and cost factors. While the capacity of the server was not a concern, the ability to send URLs and receive results at speed was, and a hired external server was deemed the most cost effective solution for the short duration of the research data gathering.

7.2.2 Data gathering

During the research period, as described in section 7.1, data was gathered on a daily basis through a script running once every twenty-four hours. This consisted of scrubbing the top ten trends on the Google research engine, and then gathering the related URLs for each trend. As covered by Moore [35] earlier, we hope to see if the changes Google has made to their ranking algorithm over the last year have done anything to stop the Search Engine Poisoning operators from being able to get results into the Google Top trends.

The decision to run the script only once every day was made for several reasons. During initial testing, running the script several times a day caused a large number of duplicate URLs which caused overhead in our processing system, as these would have to be removed, and would possibly not have given us a clear view of the amount of unique trends for a day.

While it was certainly possible to retrieve data up to once every hour, the feeling was that this would have caused more 'noise' in the data analysis than accurate data, and as such, the decision was made to run the script once a day.

Before the data was passed on to the automated tools, it was sanitised to allow us to look for and strip out duplicates or malformed URLs which the tools would be unable to process, for example, http://0.0.0.0 (a result we saw several times during the research period). These occurred for several reasons, including Google's own redirecting mechanism for pages, as well as dead links from URL shortening services such 'bit.ly'[29]. One possibility that was considered could be the use of the URL shortening services by the campaign operators as another method to disguise their campaigns. Due to the short life span of these URL's it was impossible to determine this possibility, though looking at some of the studies done by Danchev [67], Sood and Enbody [59] and Kumar [70], this is not normal behaviour in the

[29] https://bitly.com/

exploit kits and services offered.

7.2.3 Data processing

The next step consisted of taking each of the URLs and running it through an automated set of scripts which fed the URLs to various tools as described in section 3.4. These tools would then tell us if the URL had any suspicious information on it, or if it was a safe and clean site.

These automated runs took place once a week as it was decided that for some of the tools to be effective in detecting the malicious elements of a site, time needed to pass. One of the tools is fed information from other sources and as such, scanning once a week allowed the tool to be more effective in helping us detect malicious sites.

The results for the various URLs (i.e. seeing if they contain any malicious content) were then retrieved from the various tools, and all results fed into the MySQL database via various Python scripts. The beauty of some of the tools was that they even provided example scripts to interface with their API via Python, allowing for data to be submitted and retrieved much more easily.

7.2.4 Data statistics

During the 31 weeks from 1 February 2012 to 31 August 2012 that the automated data gathering ran, the total amount of non-sanitised URLs collected amounted to 44 874 URLs, with an average of 207 unique URLs per day, and 1447 URLs per week.

Figure 32 shows the amount of URLs collected every day for the research period, with small increases or decreases, depending on how well a news item or trend was picked up by high ranking websites or news sites on the internet.

Figure 33 shows the number of URLs collected every week for the same period, but allows us to show in greater detail the rise and fall of the top trends and associated sites. Two spikes in the graph show the detail of the definite increase in URLs associated with top trends, and we will be looking at that more closely later on in this chapter.

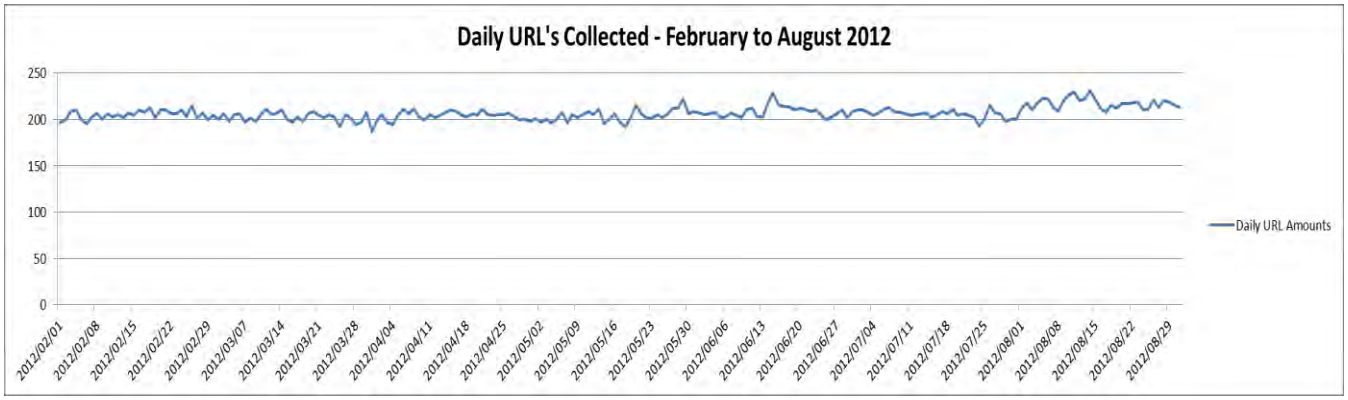

Figure 32: Daily URLs collected – February to August 2012

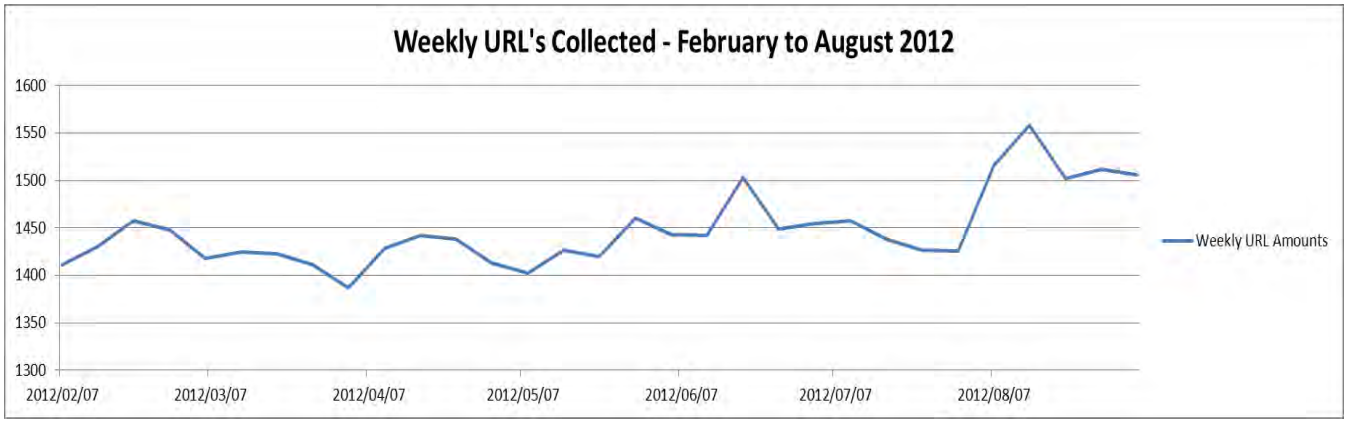

Figure 33: Weekly URLs collected – February to August 2012

7.3 Automated Data Findings

By interfacing with the VirusTotal API, as described in 3.4.2 and 7.2.3, it allowed us to classify certain URLs as either malicious or malware sites. With VirusTotal interfacing with over thirty (30) different anti-virus vendors, it gives us a good starting point for identifying websites infected with and/or used in Search Engine Poisoning campaigns.

All the collected URLs are alternatively sent to JSunpack (see 3.4.1) and Thug (3.4.3) for analysis, to identify possible malicious content. All this data is then compared and analysed to give us a fairly comprehensive picture of whether a URL is indeed suspect or not.

During our analysis, there were some false positives and errors, and we will look at some of those in section 7.3.1, and try to find an explanation for those false positives.

Once all URLs were sanitised and faulty URLs removed (as shown in section 7.2.2), the total results came to a staggering 698 943 results. The majority of these were as a result of the VirusTotal API allowing us to test a URL and retrieve multiple results from multiple sources for a single URL. During the course of our research, the resources that VirusTotal queried for results also increased, from an initial 31 to 34 sources as VirusTotal incorporated the back end databases of new tools as they were released and allowed for interfacing with their back end through an open API. This is as new research and new products are released and allow for the results to be interfaced with VirusTotal.

7.3.1 False Positives

When looking at the data on a weekly basis we did find that there were one or two incidents where there was a false positive in the data. A false positive as described by Redja [92] is "false detection or false alarm, occurs when an antivirus program detects a known virus string in an uninfected file".

In the case of our research we found that some of the engines would trigger on a word contained in a URL or on the website(s) and mark it as a malicious site. During our weekly investigations into the results of the week's processing of the data, the offending URLs were then visited to see if there was really malicious content on the website.

7.3.1.1 www.askmen.com

One of the first false positives to be identified was the site www.askmen.com. During our research from February to September 2012 this site appeared no less than sixty-seven (67) times and was always flagged as malicious by the Yandex[30] search engine. Yandex as described by its profile is one of the leading search engines and internet companies in Russia. During the research period, each of the false positives from the www.askmen.com site was investigated and nothing malicious found. The site had such a high appearance on our Google Trends as it contains up-to-date information on most celebrity topics which make news. By its own definition[31], www.askmen.com is "A leading online media and services company

[30] http://company.yandex.com/
[31] http://corp.ign.com/about/

obsessed with gaming, entertainment and everything guys enjoy". During the research period, www.askmen.com featured every time a celebrity made the Google Trends top 10 list, e.g. Christian Bale during the launch of the Batman movie 'The Dark Knight Rises', and the associated Aurora shootings[32]. Yandex search reported the site as containing the JS/Sinowal-V Trojan, being a cross-site scripting exploit. More information on cross-site scripting exploits can be found in the excellent paper by Bojinov et al [93] 'XCS: cross channel scripting and its impact on web applications'. During our investigations no evidence of this was found, even when using an out-dated and unpatched Microsoft Windows XP virtual machines (configuration A, B, C and D). During discussions with other information security practitioners the only plausible conclusion we could come to was that the site was infected at some stage in the last twelve to twenty-four months, and some of the engines still flag it as suspicious as they have not updated their records.

7.3.1.2 DNS Changer

During the week of the 9th to the 15th of July, the DNS changer malware made the headlines as the possibility existed of thousands of computers being disconnected from the internet. Baltazar et al [94] explain how the DNS changer malware infected thousands of unsuspecting computers as part of the Koobface botnet. The botnet consisted of various components, one of which was a DNS (Domain Name Server) filter / changing program that would redirect an infected computer to a malicious site, and prevent it from being able to connect to a site able to remove the software, such as an anti-virus software site.

The reason why the DNS changer malware made headlines (and thus appeared on our lists) is that on the 9th of July 2012 the FBI in the United States of America shut down two fake DNS servers operated by the Koobface gang[33].

Seven URLs relating to the DNS changer news were identified as possibly containing malicious content. During our investigations we found that certain of the sites described how the software operated including example code and thus triggered the word based and/or heuristic scanning engines of some of the sources working through the VirusTotal API.

[32] http://en.wikipedia.org/wiki/2012_Aurora_shooting
[33] http://www.pcmag.com/article2/0,2817,2406855,00.asp

These were obviously excluded from the final results, but it does show that no system for detecting malware or malicious content on the web is foolproof, and only by using multiple layers of security can a result be truly identified as malicious.

The above examples also show that in the event of vendors not updating their methods of scanning, or if they fail to repeatedly scan a URL identified as malicious, the result can easily be classified incorrectly, and thus negatively impact on the user experience of that service or of the affected URL.

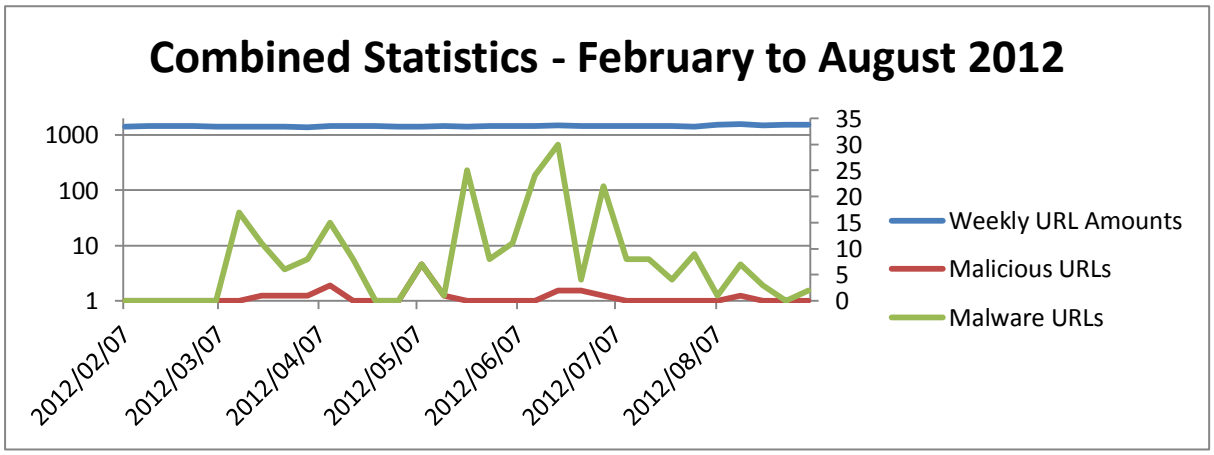

Figure 34: Combined Statistics (Logarithmic Scale) – February to August 2012

7.3.2 Malware Collection and Statistics

During the seven month period of data gathering, a total of 254 malware-containing URLs were classified by the various tools used. Of these, eight were false positives, as discussed in 7.3.1. As can be seen in Figure 31, the detected sites with anomalies (malware or malicious content) versus the total amount of scraped URLs are minuscule, with only 3.6% of all sites classified as malware. While this might not sound like a lot, the result of just one highly ranked website being compromised and serving malware to unsuspecting users can impact up to ten million users as shown by Barracuda Labs[34].

[34] http://www.barracudalabs.com/goodsitesbad/index.html

The results consisted mostly of two type of exploits, one being JS/Relink-B (as classified by Sophos) and JS/Ref-C (as classified by Sophos). These, especially the JS/Ref-C exploits, are a family of JavaScript Trojans that, when the infected page is viewed in a browser, try to redirect the user to a remote malicious site.

What was missing from the findings, something we were actively looking for, was any of the Mal/Iframe (as classified by Sophos) versions of malware. This led to another finding, the complete absence of the Black Hole Exploit Kit in the sites found, and the complete absence of fake anti-virus sites in any of the URLs that we collected. Howard [95] reports that the Black Hole Exploit Kit relies heavily on the Mal/Iframe family of injection scripts. None of these were found during the research period.

Since Brook [68] had earlier reported that the Black Hole Exploit Kit was responsible for 93% of all malicious links, the absence was investigated further, and both Giuliani [96] and Oliver et al [97] confirmed what was suspected: That the Black Hole Exploit Kit delivers its initial payload via an email containing a link to a possible exploited site, with limited attempts to gain entry into trending topics on search engines.

7.3.3 Malicious Collection and Statistics

During the research period, substantially fewer malicious results were found than malware results. In part this could be due to the classification methods of the software used, and partly due to a tool like ‘jsunpack’ being unable to distinguish between malicious javascript and malware like an adware serving toolbar.

Figure 35 represents a simple graph of the number of results classified as ‘malicious’ during our automated research. As explained above, due to the tools used the malicious classification is absent from ‘jsunpack’ but found in VirusTotal. Due to the fact that ‘jsunpack’ was tuned to look only for malicious javascript specific to iframe injections and exploit .PDFs, it was felt that it would not skew the results, and thus it was still able to give a complete picture of the nature of the collected URLs.

One of the first interesting results found in the malicious category was the prevalence of adware or malicious toolbars, and the associated downloads on sports sites. In 2005 Gordon [98] already studied the tendency of sports sites to try trick users in to downloading a toolbar which in turns spreads advertisements and spyware to users. Twenty of the results were from American football sites, e.g. http://ohiostate.scout.com/, which simply redirects to http://www.foxsportsnext.com. Looking at the WHOIS information for the site, we see it belongs to Twentieth Century Fox Film Corporation.

Registrant:

Intellectual Property Department
Twentieth Century Fox Film Corporation
P.O. Box 900
Beverly Hills CA 90213-0900
USA
domains@fox.com +1.3103691000

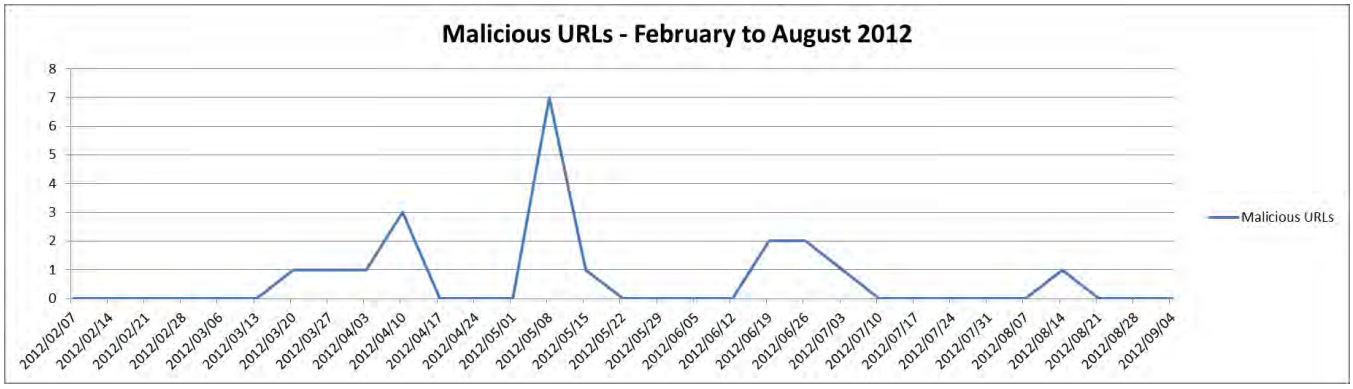

Figure 35: Malicious URLs – February to August 2012

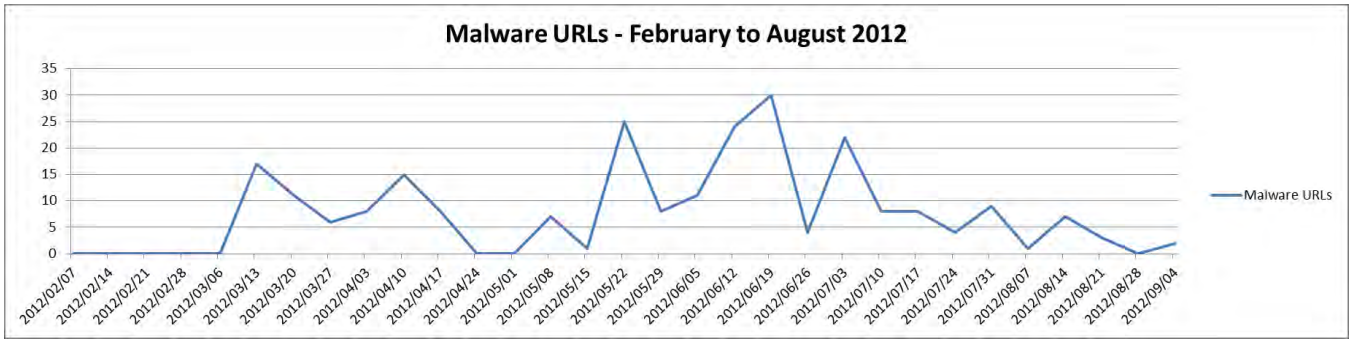

Figure 36: Malware URLs – February to August 2012

Some further research points us to reports that the site has served adware in the past, in particular Mal/Badsrc-C[35]. No reports of the actual Fox website serving malicious software were found. Further investigation showed that the chances were that adware was spread via an advertising partner running on the WordPress platform. As seen in 6.4.1 it is fairly easy for an attacker to infect a WordPress website either via vulnerability in the actual code, or via vulnerability in the plugins used in the site.

Baccas [99] does a thorough analysis of Mal/Badsrc-C, also classified as Troj/PHPShll-B, and shows how this particular exploit infects Wordpress websites, aimed specifically at users visiting the site with Internet Explorer, confirming what we saw in 6.2.2 and giving evidence to substantiate our findings. While the author is unsure of how it compromised the site, it could be anything from a compromised FTP account to a hosting provider with bad cross-user permissions — i.e. one administration account can access multiple hosted environments.

The rest of the sites on our list were entertainment and news-related sites, with no obvious pattern between them.

7.3.4 Unrated Sites Collection and Statistics

Lastly, during the period of research we had 76398 unrated results. A closer inspection of these showed that all the results were from VirusTotal. The unrated results are as a result of VirusTotal pulling data from various sources, and these sources not having information on the URLs that we had fed it. During investigation there was no consistent engine or product

[35] http://www.sophos.com/en-us/threat-center/threat-analyses/viruses-and-spyware/Mal~Badsrc-C.aspx

that filed to provide results, and no pattern could be found between the unrated results and the sites that produced this result.

Considering that we collected nearly seven hundred thousand results from VirusTotal, the unrated results constituted only one percent of the total number of results. The only concern was that in this amount of unrated results, there was the possibility of a malware site that slipped through. This was luckily offset by the fact that each result was scanned by thirty different engines, all having tied in with the Virustotal API.

7.4 Closing the loop

Any study into Search Engine Poisoning would not be complete unless a small part is spent looking into how to prevent it from happening, or affecting not just the search engine, but the end user as well. In this section we will take a look at what some of the search engines are doing, as well as what some commercial companies are doing to protect the internet user from becoming a victim of an SEP campaign.

7.4.1 Search Engines

During our manual searching efforts, as seen in Chapter 6, several instances were seen where Google had made efforts to stop sites from spreading not only malicious content, but also copyrighted information as can be seen in Figure 37.

Figure 37: Google removing a search result

This was interesting as these results were between results for other torrent-based websites which cram keywords into the search results, and offer downloads of movies and other copyright-protected content as seen in Figure 37. Thus, not only do the search engines try protect the users from harmful content but also remove entries on request from other entities. Discussing these requests is not the purpose of this thesis, thus we will not spend too much time on it, but it should be noted that it is important that search engines do attend to these requests as the website hosting the copyrighted material can and often does also engage in SEP activities.

7.4.2 Browser Safety

Another improvement during the last few years in protecting users from malicious sites, such as those serving malware through SEP attacks, is built-in protection in browsers. Certain (but not all browsers) have built in protection that warns a user should they be on a site that could exhibit malicious behaviour.

Two of the browsers that were found to successfully exhibit the ability to stop a user from visiting a known malicious site were Mozilla's Firefox and Google's Chrome browser. Without any additional add-ons or tools these browsers provided the ability to warn users as can be seen in the screenshots in Figure 38 and Figure 39.

Unfortunately this can sometimes cause some problems for sites that link from advertising partners, or partners serving content, as Mawson [100] reports. In the article, Mawson reports that http://www.iol.co.za, one of South Africa's leading news websites, was flagged for having malware. The site itself was not dangerous, but content from a third party was to blame. According to the report, it was initially the Google Chrome browser that first reported the problem to users, but the message then propagated to the search engine results provided to all users of Google, irrespective of what browser the user was using at the time.

Figure 38: Mozilla Firefox warning

Figure 39: Google Chrome warning

In such a case the site owner is then forced to contact Google and prove that the malicious code has been removed, before the warning will be removed from Google's search engine and Google Chrome's built-in warning system. While it may be an inconvenience to website owners, it does prove that some search engines are serious about protecting the end user.

7.4.3 Third party tools

Discussing the effectiveness of third party tools in stopping malware infected SEP pages would be enough to fill an entire thesis on its own. Just the search for 'safe surfing software' on Google yielded 17 400 000 results at the time of writing. Thus, we focus on the three tools used during Chapter 6 in identifying possible pages. These tools were MacAfee SiteAdvisor [101], Avast WebRep [102] and the Web of Trust [103] browser plugin. At the time of writing, the Web of Trust plugin alone had 47 744 000 downloads, giving us a good indication of the popularity (and hence effectiveness) of the tool. The following section is not a detailed discussion of the tools, but simply an overview of each of the tools in order to give the reader an effective way to combat SEP poisoned websites.

7.4.3.1 McAfee SiteAdvisor

McAfee SiteAdvisor [100] is a free application from the McAfee Corporation. The application is downloaded to the user's machine and installed. Once installed the application will add a specific warning to results and help find the user safer alternatives.

McAfee builds the reputation index by visiting websites, downloading applications where possible and testing these websites and applications for adware, spyware and viruses. Where the tools find that there is a virus, Trojan or malware included in the software or website, it flags the site as 'red' and users visiting the site are then warned.

7.4.3.2 Avast WebRep

Avast Antivirus [101] states that "Avast WebRep combines antivirus software with website ratings from millions of users in the global avast! community – providing users of avast! 6.0 with a community-sourced guide to the safety and/or content of websites. avast! users can thus know – before clicking a link – what to expect in terms of product or service quality, customer service levels, or website safety and reliability. Ratings results are "traffic-light" simple: green = GO… orange = CAUTION… red = STOP!"

Since the service depends on the input of users it can sometimes be confusing, since a site that hosts pornography might be rated as green. This is because the site itself is safe to visit,

and posts no danger to the user, but the category system that the software uses will still warn the user of the content. Another example could be a normal blog, running WordPress software, but a number of users might have had issues with the site serving fake anti-virus, and thus rated the site as dangerous as per the example we found in Chapter 6.

7.4.3.3 Web of Trust

Web of Trust [102] is another community driven project, using a reputation database to guide users. The application itself is an internet browser add-on that installs locally on the user's computer, and once installed, gives a website rating for every site that a user might visit, based on the reputation database.

The reputation database is built on user submissions, as well as submissions from websites like phishtank[36]. It also uses a simple green, orange and red classification scheme to warn users. Unlike the Avast! categories, Web of Trust uses only four categories, one of which is 'child safety'. Thus a website that might contain pornography can get a green rating for trustworthiness, privacy, and vendor reliability, but get a red rating for child safety.

7.5 Summary

This chapter has provided an overview of our automated research dataset, looking at the data that was collected in the 31 week period from February to August 2012. The chapter looked at how the data was processed, the amount of URLs contained in the dataset, the amount of results that were processed, and the malware and malicious URLs found in the dataset. The research results were positive with a small amount of malware and malicious URLs found. These were analysed and the malware or malicious content identified.

A small section was also dedicated to the false positives in the dataset and the reasons or causes of these false positives, including investigating and analysing some of the more specific URLs that were flagged as false positives.

[36] http://www.phishtank.com/

Lastly time was spent looking at ways that are being implemented by search engines, open source projects and commercial products to try to prevent Search Engine Poisoning campaigns from exploiting and infecting users on the internet, including example outputs from these sources.

This chapter concludes the analysis done for the data sets that were collected during the research period. The following chapter contains the conclusions and summary for the thesis, where the research objectives are revisited and possibilities for future research are looked at.

8.Conclusion

The research that comprises this thesis consists of two sets of findings and a historical view of Search Engine Poisoning. The historical view in Chapter 4 looks at two celebrity deaths and the associated Search Engine Poisoning campaigns that were launched with those deaths and a third campaign that focused on a sexual content and thus peaked user curiosity through such means to try lure victims to the campaign.

In Chapter 6 and Chapter 7 the results of a manual and an automated search process are discussed with several campaigns found, spread through different means, including compromised plugins in popular web hosting software. The two sets of findings are as follows:

- A manual set of three search results and the associated data analysis that was done with these results. These results were collected between June 2012 and July 2012 and are discussed at length in Chapter 6.

- A dataset of 44 874 URLs presented in Chapter 7, with a combined result of 1 346 220 results attained through the various toolsets used (as discussed in Section 3.4), including 698 943 alone from the VirusTotal API. This was collected from 1 February 2012 to 31 August 2012 over a 31 week period. Of this collection 254 were identified as possibly having malware content, and a further 20 malicious URLs were found.

8.1 Research objectives

As defined in chapter 1, our research objectives were as follows:

1. Firstly, to look at and investigate case studies of Search Engine Poisoning in 2011, particularly with regards to news events that made headlines. This was done to

establish a baseline for the type of results that we hoped to find in the manual and automated research chapters. Several examples were found and explored in detail in Chapter 4.

2. The second objective of the research was to look for Search Engine Poisoned sites using manual search methods. During this period of research several examples of Search Engine Poisoning were found with relative ease, and investigated in detail. The results were compared to historic data from academic and industry sources, and allowed us to make comparisons to determine behaviour changes in techniques and results, along with proving just how easily these campaigns still catch the average internet user.

3. The third part of the research involved retrieving data over a period of seven months and running this collected data through various tools as described in Chapter 3. The collected data was then analysed for trends and the results presented.

8.2 Research Findings

Our research findings have been split into two parts: The manual searching, and the automated searching results.

8.2.1 Manual Research Findings

The manual research imitated the behaviour of the average internet user, searching for information on trending topics, celebrities, or news events, and provided us with three different results from three different Search Engine Poisoning campaigns, each delivering a different type of malware to the victim's computer. The first result delivered a botnet Trojan to the test machine, and the next two results netted the research a fake anti-virus program and an ad-ware installer.

These results were obtained searching for simple keywords and focusing on some trending topics at the time, including the 2012 Summer Olympics. These results simply proved that

Search Engine Poisoning campaigns are still running on the internet, that these campaigns are taking on different approaches in the way they try to lure the user, and that they deliver various different types of malware to the victim's computer. We can safely assume that the industry concern over Search Engine Poisoning is legitimate, and all internet users in the foreseeable future would be wise to exercise caution.

8.2.2 Automated Research Findings

The automated research in this paper focused on the gathering of the top ten trending topics on the Google search engine, gathering the URLs associated with these trends on a daily basis, and testing each of these URLs in our dataset for Search Engine Poisoning content through the use of three different tools.

Our total result set comprised of 1 346 220 results over the research period from the beginning of February to the end of August 2012 (a period of 31 weeks), obtained from 44 874 URLs. The results from this dataset included 254 URLs that contained some sort of malware, and a further 20 malicious URLs. There were a total of 76398 unrated results in our dataset, but since the collected data was run through multiple sources, this was considered not to have skewed the results.

The false positives that were identified and analysed led to an interesting discovery: That some of the engines we used in the scanning keep residual data for an extended period of time, and in doing so, mark sites that are now clean and safe as still dangerous.

Looking at the complete dataset that was analysed, it still proves that Search Engine Poisoning does make its way into the top trending topics on search engines — and not just through obscure results, but in the top 200 URLs that typical users might visit during the course of a day.

8.3 Future Research

Three possible objectives of future research have come to the fore.

1. Applying the same automated search mechanism for the .za TLD, focusing specifically on South Africa, and attacks that are aimed at, or originate from sites in the TLD. To date, no such research has been found, but it could very well be that there is such research in progress.

2. Adapting the research methodology and identifying the Search Engine Poisoning landscape within the African continent. During the Arab Spring, and the death of Gaddafi, several Search Engine Poisoning campaigns were seen on the internet, and thus looking at a specific African context should yield some interesting results.

3. Studying the effectiveness of third party software in combating malicious websites, and comparing gateway efforts versus end point efforts through a set collection of malicious sites. An addition to this study would include looking at the effectiveness of commercial versus open source products.

8.4 Conclusion

The research objective during this thesis, and the various methods of research employed during the gathering of the datasets used (both manual and automated), set out to look at the prevalence of Search Engine Poisoning in modern search engines.

Research in this thesis looked at the evolution of exploit kits on the internet, how they evolved, and the current eco-system around Search Engine Poisoning and how easy it is for criminals to run a campaign.

Three examples of S;earch Engine Poisoning were examined during the manual data gathering, and 254 results reporting malware were found during our 31 week automated data gathering period.

The final chapter showed that even with all the defensive measures in place by search engines, they are still exploited by Search Engine Poisoning campaigns and that it is still a caution for the everyday internet user.

References

[1] Jonathan Lynn, David Stamp, and Stephanie Nebehay. (2010, October) Reuters. [Online]. http://www.reuters.com/article/2010/10/19/us-telecoms-internet-idUSTRE69I24720101019 - Accessed 5 December 2012

[2] G. Hotchkiss, S. Alston, and G. Edwards, "Eye tracking study," *Research white paper, Enquiro Search Solutions Inc*, 2005.

[3] P.A. Hamilton, Google-bombing-Manipulating the PageRank Algorithm, 2009, http://userpages.umbc.edu/~pete5/ir_paper.pdf.

[4] Dean, J., "Challenges in building large-scale information retrieval systems: invited talk," in *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09), Ricardo Baeza-Yates, Paolo Boldi, Berthier Ribeiro-Neto, and B. Barla Cambazoglu (Eds.)* , vol. DOI=10.1145/1498759.1498761 http://doi.acm.org/10.1145/1498759.1498761, New York, NY USA, 2009, pp. 1-1.

[5] Clifford Tatum, "Deconstructing Google bombs: A breach of symbolic power or just a goofy prank?," *First Monday*, vol. 10, no. 10, 2005.

[6] Stephanie Buck. (2012, April) Mashable. [Online]. http://mashable.com/2012/04/19/google-bombs/ - Accessed 4 December 2012

[7] T. McNichol. (2004, Jan) NY Times - Engineering Google results to make a point. [Online]. http://cs.wellesley.edu/~cs315/Papers/NYT-miserableFailure.pdf - Accessed 6 December 2012

[8] K Bryan and T. Leise, "The $25,000,000,000 eigenvector: The linear algebra behind Google," *Society for Industrial and Applied Mathematics review*, vol. 48, no. 3, pp. 569-581, 2006.

[9] J. Bar-Ilan, "Google bombing from a time perspective," *Journal of Computer-Mediated Communication*, vol. 12, no. 3, pp. 910-938, 2007.

[10] P. Hayati and V. Potdar, "Spammer and hacker, two old friends," in *In Digital Ecosystems and Technologies, 2009. DEST'09. 3rd IEEE International Conference on*, 2009, pp. 290-294.

[11] Z. Gyongyi and H. Garcia-Molina, "Web spam taxonomy," in *First International*

Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005), Chiba, Japan, 2005, pp. 39-47.

[12] G. Mishne, D. Carmel, and R. Lempel, "Blocking blog spam with language model disagreement," in *AIRWeb'05: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, 2005, pp. 1-6.

[13] Jeff Yan, "Bot, cyborg and automated turing test," in *Security Protocols*, Bruce Christianson et al., Eds. Manchester, United Kingdom: Springer, 2009, pp. 190-197.

[14] A. Thomason, "Blog spam: A review," in *Proceedings of Conference on Email and Anti-Spam (CEAS)*, 2007.

[15] J. Caverlee, M. Srivatsa, and L. Liu, "Countering Web Spam Using Link-Based Analysis," in *2nd Cyber Security and Information Infrastructure Research Workshop (CSIIRW)*, Oak Ridge National Laboratory, 2006.

[16] Y.M. Wang, D. Beck, X. Jiang, and R. Roussev, "Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities," in *Proceedings of the 2006 Network and Distributed System Security Symposium*, 2006, pp. 35-49.

[17] J.P. John, F. Yu, Y. Xie, A. Krishnamurthy, and M. Abadi, "deSEO: Combating Search-Result Poisoning," in *Proc. USENIX Security. Vol. 11.*, 2011.

[18] N. Leontiadis, T. Moore, and N. Christin, "Measuring and analyzing search-redirection attacks in the illicit online prescription drug trade," in *Proceedings of USENIX Security*, 2011, pp. 3-20.

[19] F. Howard and O. Komili, Poisoned search results: How hackers have automated search engine poisoning attacks to distribute malware, Sophos Technical Papers, 2010.

[20] J. Caballero, C. Grier, C. Kreibich, and V. Paxson, "Measuring Pay-per-Install: The Commoditization of Malware Distribution," *Proc. of the USENIX Security*, 2011.

[21] Ed Bott, Bing ad serves malware to would-be Google Chrome switchers, August 2011, http://www.zdnet.com/blog/bott/bing-ad-serves-malware-to-would-be-google-chrome-switchers/3687 - Accessed 6 December 2012.

[22] Ed Bott, Bing ads lead to more malware; new Mac Trojan in the wild, August 2011, http://www.zdnet.com/blog/bott/bing-ads-lead-to-more-malware-new-mac-trojan-

in-the-wild/3702 - Accessed 6 December 2012.

[23] Ed Bott, Social engineering in action: how web ads can lead to malware, July 2011, http://www.zdnet.com/blog/bott/social-engineering-in-action-how-web-ads-can-lead-to-malware/3532 - Accessed 6 December 2012.

[24] J. Newsome, B. Karp, and D. Song, "Polygraph: Automatically generating signatures for polymorphic worms," in *Security and Privacy, 2005 IEEE Symposium on. IEEE*, 2005, pp. 226-241.

[25] Tom Clare, Blue Coat 2011 Web Security Report, 2011, http://www.zdnet.com/blog/bott/bing-ads-lead-to-more-malware-new-mac-trojan-in-the-wild/3702 - Accessed 6 December 2012.

[26] Chris Larsen, Latest SEP (Search Engine Poisoning) Research, Part 3, March 2012, http://www.bluecoat.com/security/security-archive/2012-03-07/latest-sep-search-engine-poisoning-research-part-3 - Accessed 6 December 2012.

[27] Chris Larsen, Latest SEP (Search Engine Poisoning) Research, Part 1, February 2012, http://www.bluecoat.com/security/security-archive/2012-02-15/latest-sep-search-engine-poisoning-research-part-1 - Accessed 6 December 2012.

[28] M.A. Rajab, L. Ballard, P. Mavrommatis, N. Provos, and X. Zhao, "The nocebo effect on the web: an analysis of fake anti-virus distribution," in *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2010.

[29] Lionic Corporation. (2011, March) Lionic Corporation. [Online]. http://www.lionic.com/userfiles/Security/2011331163056.pdf

[30] Dennis Fisher, Malicious Ads on Bing Lead to ZeroAccess Trojan, October 2011, http://threatpost.com/en_us/blogs/malicious-ads-bing-lead-zeroaccess-trojan-101411 - Accessed 6 December 2012.

[31] Kelly Clay, Emma Watson Tops List of Most Dangerous Celebrities, September 2012, http://www.forbes.com/sites/kellyclay/2012/09/11/emma-watson-tops-list-of-most-dangerous-celebrities/ - Accessed 6 December 2012.

[32] John Leydon, Bing is the most heavily poisoned search engine, study says, October 2012, http://www.theregister.co.uk/2012/10/08/bing_worst_search_poisoning/ - Accessed 6 December 2012.

[33] Kim Eichorn. (2011, September) McAfee. [Online]. http://www.mcafee.com/us/about/news/2011/q3/20110915-02.aspx - Accessed 6 December 2012

[34] Grayson Lenik. (2012, August) An Eye On Forensics. [Online]. http://eyeonforensics.blogspot.com/2010/08/defcon-18.html - Accessed 6 December 2012

[35] T. Moore, N. Leontiadis, and N. Christin, "Fashion crimes: trending-term exploitation on the web," in *Proceedings of the 18th ACM conference on Computer and communications security. ACM*, 2011, pp. 455-466.

[36] B. Hartstein, "Jsunpack: An Automatic JavaScript Unpacker," in *ShmooCon convention*, Washington, D.C., 2009.

[37] Angelo Dell'Aera. (2011) Thug. [Online]. http://buffer.github.com/thug/ - Accessed 6 December 2012

[38] M.T. Qassrawi and H. Zhang, "Detecting malicious web servers with honeyclients," *Journal of Networks*, vol. 6, no. 1, pp. 145-152, 2011.

[39] C. Seifert, I. Welch, P. Komisarczuk, and others, "Honeyc-the low-interaction client honeypot," *Proceedings of the 2007 New Zealand Computer Science Research Student Conference, Waikato University, Hamilton, New Zealand*, 2007.

[40] K.F. Durkin, "Death, dying, and the dead in popular culture," *Handbook of death and dying*, pp. 43-49, 2003.

[41] K. Katzman, "Al Qaeda: Profile and threat assessment," in *Library Of Congress Washington DC Ccongressional Research Service*, August 2005.

[42] M.J. Langworthy, J. Sabra, and M. Gould, "Terrorism and blast phenomena: lessons learned from the attack on the USS Cole (DDG67)," *Clinical orthopaedics and related research*, vol. 422, p. 82, 2004.

[43] O. Laden, B. Lawrence, and J. Howarth, *Messages to the World: The Statements of Osama bin Laden*.: Verso - ISBN 1844670457, 2005, Translated by James Howarth. London: Verso.

[44] Wally Bock, Lessons from September 11, September 2002,

http://agreatsupervisor.com/articles/911lessons.htm - Accessed 6 December 2012.

[45] M. Mazzetti, H. Cooper, and P. Baker, "Behind the hunt for bin Laden," *The New York Times*, May 2 2011.

[46] Fabio Assolini. (2011, May) http://www.securelist.com/. [Online]. http://www.securelist.com/en/blog/6202/Blackhat_SEO_and_Osama_Bin_Laden_s_death - Accessed 6 December 2012

[47] Graham Cluley, Osama bin Laden death video scam spreads virally on Facebook, May 2011, http://nakedsecurity.sophos.com/2011/05/02/osama-bin-laden-death-video-scam-spreads-virally-on-facebook/ - Accessed 6 December 2012.

[48] Ben Sisario, "Amy Winehouse, a troubled star gone too soon," *Qatar Tribune*, p. 1, June 2011.

[49] R.L. Shaw, C. Whitehead, and D.C. Giles, "'Crack down on the celebrity junkies': Does media coverage of celebrity drug use pose a risk to young people?," *Health, Risk & Society - Taylor & Francis - ISSN 1369-8575*, vol. 12, no. 6, pp. 575-589, December 2010.

[50] A. McRobbie, "Pornographic permutations," *The Communication Review*, vol. 11, no. 3, pp. 225-236, 2008.

[51] A.E.L. Kory, *In Memory of Amy Winehouse*.: Lulu. com, 2011.

[52] Cris Lumague, Amy Winehouse's Death Used in Online Attacks, July 2011, http://blog.trendmicro.com/amy-winehouses-death-used-in-online-attacks/ - Accessed 6 December 2012.

[53] Steve Ragan, Amy Winehouse scams jump from Facebook to email, July 2011, http://www.thetechherald.com/articles/Amy-Winehouse-scams-jump-from-Facebook-to-email - Accessed 6 December 2012.

[54] Nick Pisa, Angry parents withdrew pupil from prestigious Milan school because teacher 'was too sexy', October 2010, http://www.dailymail.co.uk/news/article-1322497/Ileana-Tacconelli-sexy-Parents-withdrew-pupil-Milan-Catholic-School.html - Accessed 6 December 2012.

[55] Staff Reporter. (2011, January) The Sun. [Online]. http://www.thesun.co.uk/sol/homepage/news/3190742/Storm-over-too-sexy-

teacher.html - Accessed 6 December 2012

[56] Crescencio Reyes. (2010, December) PC Tools. [Online]. http://www.pctools.com/security-news/ileana-tacconelli-fake-adobe-flash-update-tdss/ - Accessed 6 December 2012

[57] Dan Goodin, 'Indestructible' rootkit enslaves 4.5m PCs in 3 months', June 2011, http://www.theregister.co.uk/2011/06/29/tdss_alureon_advances/ - Accessed 6 December 2012.

[58] N. Villeneuve, "Targeting The Source," 2011, http://www.trendmicro.co.uk/media/wp/fakeav-affiliate-networks-whitepaper-en.pdf.

[59] Aditya Sood and Richard Enbody, Browser Exploit Packs – Exploitation Tactics, October 2011, Virus Bulletin Conference October 2011.

[60] Dancho Danchev, DIY Exploits Embedding Tools - a Retrospective, September 2007, http://ddanchev.blogspot.com/2007/09/diy-exploits-embedding-tools.html - Accessed 6 December 2012.

[61] Chris Boyd. (2007, September) SpywareGuide. [Online]. http://blog.spywareguide.com/2007/09/compromised_emails_lead_to_ie.html - Accessed 6 December 2012

[62] Peter Coogan. (2010, February) Symantec. [Online]. http://www.symantec.com/connect/blogs/fragus-exploit-kit-changes-business-model - Accessed 6 December 2012

[63] Daniel Chechik. (January, 2011) M86 Security Labs. [Online]. http://labs.m86security.com/tag/exploit-kit/ - Accessed 6 December 2012

[64] Artem Gololobov. (2011, March) Websense. [Online]. http://community.websense.com/blogs/securitylabs/pages/crimepack-exploit-kit.aspx - Accessed 6 December 2012

[65] Daniel Chechik. (2012, March) M86 Security Labs. [Online]. http://labs.m86security.com/tag/phoenix-exploit-kit/ - Accessed 6 December 2012

[66] Brian Krebs. (2011, January) Krebs on Security. [Online]. http://krebsonsecurity.com/2011/01/exploit-packs-run-on-java-juice/ - Accessed 6

December 2012

[67] Dancho Danchev, Cybercriminals release 'Sweet Orange' – new web malware exploitation kit, May 2012, http://blog.webroot.com/2012/05/10/cybercriminals-release-sweet-orange-new-web-malware-exploitation-kit/ - Accessed 6 December 2012.

[68] Christopher Brook, Blackhole Exploit Kit's Dominance On Infected Hosts Could Push Rivals To the Cloud, February 2012, http://threatpost.com/en_us/blogs/blackhole-exploit-kits-dominance-infected-hosts-could-push-rivals-cloud-020812 - Accessed 6 December 2012.

[69] Jason Jones. (2012, July) Black Hat 12. [Online]. http://media.blackhat.com/bh-us-12/Briefings/Jones/BH_US_12_Jones_State_Web_Exploits_Slides.pdf - Accessed 6 December 2012

[70] Mohit Kumar. (2011, May) The Hacker News. [Online]. http://thehackernews.com/2011/05/blackhole-exploit-kit-download.html - Accessed 6 December 2012

[71] Artem Gololobov. (2011, February) Websense. [Online]. http://community.websense.com/blogs/securitylabs/pages/black-hole-exploit-kit.aspx - Accessed 6 December 2012

[72] Pradeep Kulkarni. (2012, January) Zscaler Research. [Online]. http://research.zscaler.com/2012/01/popularity-of-exploit-kits-leading-to.html - Accessed 6 December 2012

[73] Kevin F. King, "Geolocation and Federalism on the Internet: Cutting Internet Gambling's Gordian Knot," *Columbia Science and Technology Law Review*, vol. 11, pp. 41-59, 2012.

[74] Chris Astacio, Phoenix Exploit's Kit, January 2011, http://community.websense.com/blogs/securitylabs/pages/phoenix-exploit-s-kit.aspx - Accessed 6 December 2012.

[75] abuse.ch. (2012, March) abuse.ch. [Online]. http://www.abuse.ch/?p=3658 - Accessed 6 December 2012

[76] Antti Tikkanen. (2012, April) F-Secure. [Online]. http://www.f-

secure.com/weblog/archives/00002344.html - Accessed 8 December 2012

[77] J. Zhang, C. Yang, Z. Xu, and G. Gu, "PoisonAmplifier: A Guided Approach of Discovering Compromised Websites through Reversing Search Poisoning Attacks," in *15th International Symposium on Research in Attacks, Intrusions and Defenses (RAID'12), September 12-14, 2012*, Vrije Universiteit, Amsterdam, The Netherlands, 2012.

[78] Gregg Keizer. (2012, October) Computerworld. [Online]. http://www.computerworld.com/s/article/9232879/Windows_XP_turns_11_still_not_dead_yet

[79] McAfee, ZeroAccess Rootkit, July 2012, https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/23000/PD23412/en_US/McAfee%20Labs%20Threat%20Advisory-ZeroAccess.pdf.

[80] Brian Krebs, "Host of Internet spam groups is cut off," *Washington Post, Nov*, November 2008.

[81] Technical Editor. (2010, September) The Official Microsoft Blog. [Online]. http://blogs.technet.com/b/microsoft_blog/archive/2010/09/08/r-i-p-waledac-undoing-the-damage-of-a-botnet.aspx - Accessed 6 December 2012

[82] Technical Editor. (2011, September) The Official Microsoft Blog. [Online]. http://blogs.technet.com/b/microsoft_blog/archive/2011/09/27/microsoft-neutralizes-kelihos-botnet-names-defendant-in-case.aspx - Accessed 6 December 2012

[83] Technical Editor. (2011, 22 September) The Official Microsoft Blog. [Online]. http://blogs.technet.com/b/microsoft_blog/archive/2011/09/22/rustock-civil-case-closed-microsoft-refers-criminal-evidence-to-fbi.aspx - Accessed 6 December 2012

[84] Den Clements. (2012, January) keyboardcrime.com. [Online]. http://keyboardcrime.com/microsoft-and-russian-spammers-what-do-they-have-in-common-part-2/ - Accessed 6 December 2012

[85] David Dede, WordPress plugins hacked – Understanding the backdoor, June 2011, http://blog.sucuri.net/2011/06/wordpress-plugins-hacked-understanding-the-

backdoor.html - Accessed 6 December 2012.

[86] Regina Smola. (2012, June) WPSecurityLock. [Online]. http://www.wpsecuritylock.com/wordpress-plugin-vulnerabilities-and-fixes-06-12-2012/ - Accessed 8 December 2012

[87] Kaspersky Lab. (2012, August) Kaspersky Lab Technical Support. [Online]. http://support.kaspersky.com/viruses/rogue?page=3&qid=208287158 - Accessed 6 December 2012

[88] Privacy Protect. Privacy Protect. [Online]. http://privacyprotect.org/about-privacyprotection/ - Accessed 6 December 2012

[89] Dan Haywood, Olympics hit by SEO poisoining, as black hat hackers change tactics, August 2012, http://www.scmagazineuk.com/olympics-hit-by-seo-poisoining-as-black-hat-hackers-change-tactics/article/253088/ - Accessed 6 December 2012.

[90] Jennifer Waters, Olympics websites may hijack your computer, July 2012, http://www.marketwatch.com/story/olympics-websites-may-hijack-your-computer-2012-07-25 - Accessed 6 December 2012.

[91] C. Gutzman, S. Sweep, and A. Tambo, "Differences and similarities of spyware and adware," *University of Minnesota Morris*, 2003.

[92] Randy Redja, What is a false positive?, February 2006, http://service1.symantec.com/sarc/sarc.nsf/info/html/what.false.positive.html - Accessed 6 December 2012.

[93] H. Bojinov, E. Bursztein, and D. Boneh, "XCS: cross channel scripting and its impact on web applications," in *ACM*, 2009, pp. 420-431.

[94] J. Baltazar, J. Costoya, and R. Flores, "The real face of Koobface: The largest web 2.0 botnet explained," *Trend Micro Research*, vol. 5, no. 9, p. 10, 2009.

[95] Fraser Howard, Exploring the Blackhole exploit kit, March 2012, http://sophosnews.files.wordpress.com/2012/03/blackhole_paper_mar2012.pdf.

[96] Gianluca Giuliani, Breaking News: The malicious USA Presidential spam campaign has started., October 2012, http://community.websense.com/blogs/securitylabs/archive/2012/10/10/breaking-news-the-malicious-usa-presidential-spam-campaign-started.aspx - Accessed 6

December 2012.

[97] Jon Olivier et al., Blackhole Exploit Kit: A Spam Campaign, Not a Series of Individual Spam Runs, 2012, http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_blackhole-exploit-kit.pdf.

[98] Sarah Gordon, "Fighting spyware and adware in the enterprise.," *The EDP Audit, Control and Security Newsletter (EDPACS)*, vol. 32, no. 12, pp. 14-18, 2005.

[99] Paul Baccas, Troj/PHPShll-B: Malware injects itself into WordPress installations, September 2011, http://nakedsecurity.sophos.com/2011/09/19/malware-wordpress-installations/ - Accessed 6 December 2012.

[100] Nicola Mawson, IOL site waiting for Google, July 2012, http://www.itweb.co.za/index.php?option=com_content&view=article&id=56603:iol-site-waiting-for-google&catid=147 - Accessed 6 December 2012.

[101] John McAfee. McAfee SiteAdvisor. [Online]. http://www.siteadvisor.com/howitworks/index.html - Accessed 6 December 2012

[102] Avast Software. Avast! [Online]. http://www.avast.com/index - Accessed 6 December 2012

[103] WOT Services Oy. Web Of Trust. [Online]. http://www.mywot.com/ - Accessed 6 December 2012