A Novel Approach of Filtration for Analysis and Detection of Malicious Web Pages

Mr.Pranav More[1], Prof. Amruta Amune[2]

Dept. of Computer Networks, G. H. Raisoni COEM (SPPU),
Ahemadnagar, India

[1]pranavmore2530@gmail.com
[2]amruta.amune@raisoni.net

Abstract - Internet services are more necessary part of our daily life. We rely growingly on the ease and flexibility of Internet connected devices to shop, communicate and in general perform tasks that would require our physical presence. While very valuable, Internet transactions can represent user sensitive information. Banking sector's and personal medical records, system authorization passwords and personal communication records can easily become known to an enemy who can easily compromise any of the devices include in online transactions. Regrettably, In this transaction the user's personal computer seems to be the weakest link. At the same time attacker also use new attacks for identification of user's sensitive information with vulnerabilities that use a small part of code in web pages. Overcome these problems use a novel approach for a filtering technique to finding malicious web pages very effectively and efficiently using supervised machine learning. Also detailed study some other techniques researcher research to analysis and detection of malicious web pages.

Keywords — *Malicious web page analysis & classification, download exploits, pronging, web page filtering.*

*****

I. Introduction

Today's internet has become an fundamental part in the life of hundreds of millions of people who usually use online services to accumulate and manage delicate information. At the same time attacker also search a new way or number of attacks to harm user's delicate or important information. Using that information they target vulnerability in a web browser.

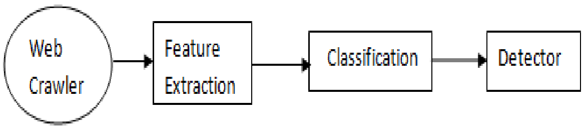

Fig. 1.1: System Overview[1]

In this paper they present a filtering technique for detect malicious web pages called profiler. In this technique use web crawler for detecting URL for browser, after that using feature extraction pages divide including their set of feature like HTML code, Java Script code etc. ,then classify that pages using supervised machine learning algorithm and finally detect malicious web page[1].

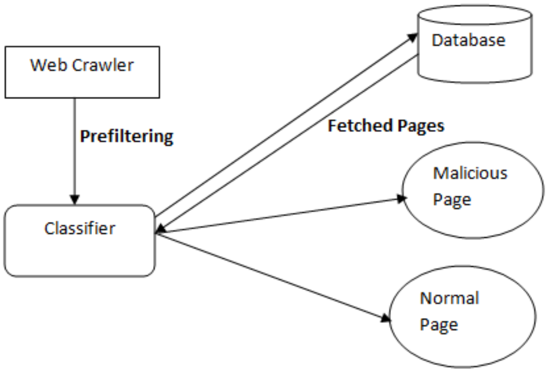

Fig. 1.2: Architecture of the system [1]

In this system classifier is heart of architecture. Using classifier classify the pages at that time they consider stop words, matching words, steaming and pronging etc. using all that things system distribute or classify malicious pages and normal pages that not damaged to the system.

II. Literature Survey

Related to this system researchers research many system and some drawback, problems are arises. Niels Provos et al. they described a number of server- and client-side exploitation techniques that are used to spread malware, and elucidated the mechanisms by which a successful exploitation chain can start and continue to the automatic installation of malware.

As well as they described present a detailed analysis of the malware serving infrastructure on the web using a large amount of malicious URLs. Using this data, they estimate the global prevalence of drive-by downloads, and identify several trends for different aspects of the web malware problem and some analysis of distribution sits and network.

Fig.2.1 shows typical interaction that takes place when a user visits a website with injected malicious content. Upon visiting this website, the browser downloads the initial exploit script. The exploit script (in most cases, JavaScript) targets vulnerability in the browser or one of its plugins. Successful exploitation of one of these vulnerabilities results in the automatic execution of the exploit code, thereby triggering a drive-by download.

Our study uses a large scale data collection infrastructure that continuously detects and monitors the behavior of websites that perpetrate drive-by downloads. Our in-depthanalysis of over 66 million URLs (spanning a 10 month period) reveals that the scope of the problem is significant. For instance, we find that 1.3% of the incoming search queries to Google's search engine return at least one link to a malicious site. Their results, on roughly 17, 000 URLs, showed that about 200 of these were dangerous to users [2].

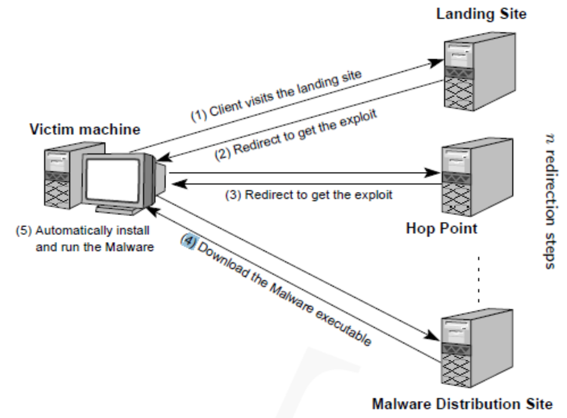
Landing Site
Victim machine
(1) Client visits the landing site
(2) Redirect to get the exploit
(3) Redirect to get the exploit
Hop Point
n redirection steps
(5) Automatically install and run the Malware
(4) Download the Malware executable
Malware Distribution Site

Fig. 2.1: A typical Interaction with of drive-by download victim with a landing URL.

Marco cova et al. described that they design detection and analysis of malicious JavaScript code using this approach combines irregular detection with emulation to automatically identify malicious Java- Script code and to support its analysis. They implement a novel approach to the automatic detection and analysis of malicious web pages. For this, we visit web pages with an instrumented browser and record events that occur during the interpretation of HTML elements and the execution of JavaScript code. For each event (e.g., the instantiation of an ActiveX control via JavaScript code or the retrieval of an external resourcevia an iframe tag), we extract one or more features whose values are evaluated using anomaly detection techniques. Anomalous features allow us to identify malicious content even in the case of previously-unseen attacks. Our features are comprehensive and model many properties that capture intrinsic characteristics of attacks. Moreover, our system provides additional details about the attack. For example, it identifies the exploits that are used and the Obfuscated version of the code, which are helpful to explain how the attack was executed and for performing additional analysis. We implemented our approach in a tool called JSAND (JavaScript Anomaly-based analysis and Detection), and validated it on over 140,000 web pages.

Using seven datasets the results of the evaluation shows that it is possible to reliably detect malicious code by its behavior of the code and comparing this behavior with a normal Java-Script execution [3]. In summary, our results indicate that JSAND achieves A Detection rate that is significantly better than state-of-the-art tools. It detected a large number of malicious pages not detected by other tools, and it missed only a very limited number of attacks. Furthermore, in our tests, JSAND analyzed about two samples per minute, which is faster than other tools. Since the analysis can be easily parallelized, performance can be further improved.

John P. John et al. described that they present SearchAudit, a suspicious- query generation framework that identifies malicious queries by auditing search engine logs. While auditing is an important component of system security. SearchAudit is a framework that identifies malicious queries from massive search engine logs in order to uncover their relationship with potential attacks. SearchAudit takes in a small set of malicious queries as seed, expands the set using search logs, and generates regular expressions for detecting new malicious queries.

Working with SearchAudit consists of two stages:-

1. Identification: - SearchAudit identifies malicious queries.
2. Investigation:- With Search Audits assistance, they focus on analyzing those queries and the attacks of which they are part.The analysis shows that the identification of malicious searches can help detect and prevent large-scale attacks [4].

Overall, SearchAudit identifies over 40,000 IPs issuing more than 4 million malicious queries, resulting in over 17 million page views.

III. IMPLEMENTATION DETAILS

A. System Architecture.

The features extraction from web pages is depend on if a web page is malicious or not. Using dynamic approaches it is possible to perform the analysis faster. The feature extraction of pages basically works on two main source of information such as the page content and the page URL.

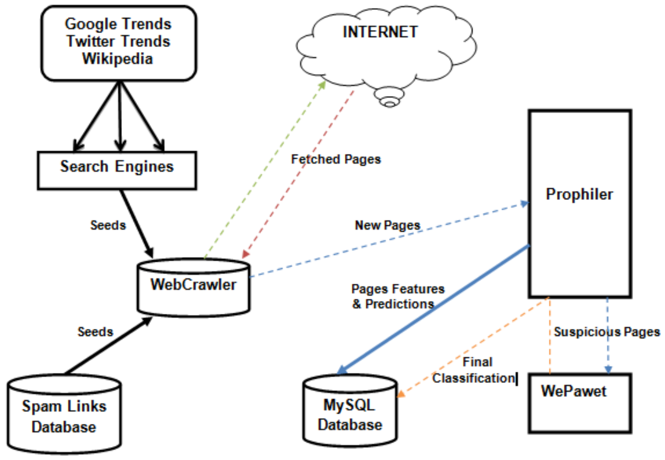

Fig.3. System Architecture

The complete system architecture is shown in figure 1. Major components of the projected system are as follows:

- **WebCrawler:** Web crawler fetch a list of seed URLs fetched daily from search engines such as Google, Yahoo and Bing etc. Web crawler also extracts a list of links from a spam emails. The list of links is updated daily. Some of the malicious web pages execute malicious content only when the request is made. The malicious contents are appears when a user clicking on the search results. This header has to be set to the search engine from which the seed URL was extracted.

- **Prophiler:** The extracted pages extracted by web crawler will submit to the Prophiler. Prophiler analyzes each page and extracts all the features of pages and stores them. Prophiler uses the models to evaluate its maliciousness on the basis of features extracted by an algorithm.

- **Wepawet:** The page is forwarded to the analysis tool known as Wepawet If a page has been identified malicious. Wepawet then confirms that the page is indeed malicious.

B. *Feature Extraction*

The features extraction from web pages is depend on if a web page is malicious or not. Using dynamic approaches it is possible to perform the analysis faster. The feature extraction of pages basically works on two main source of information such as the page content and the page URL. Using that type of information use to extract the some features such as:

- **HTML Features:**

This feature is based on statistical information including the raw content of a page and on structural information that derived from parsing the HTML code. To parse HTML pages using Neko HTML Parser because of its versatility and speed in parsing HTML code. Since some of below features detect anomalies in the structure of a web page, which are silently corrected by Neko, as well as using parse web pages with HTMLparser, which performs no error Correction. The some of features are as follows:

1. Number of elements with small area.
2. Number of elements containing suspicious content.
3. Number of suspicious objects.
4. Number of included URLs.
5. Number of out of place elements.
6. Presence of double documents.
7. Number of known malicious patterns

- **JAVA SCRIPT Features**:

This type of features result based on static analysis. Similarly to HTML features, JavaScript features are both statistical and lexical. Most malicious JavaScript scripts are difficult analysis time. In some cases, malware authors adopt encryption scheme and techniques to prevent code debugging. This implemented the extraction of some statistical measures. It also considers the structure of the Java Script code itself. A number of features is based on the analysis of the Abstract Syntax Tree (AST) extracted using the parser. The some of features are as follows:

1. Keywords-to-words ratio.
2. Number of long strings
3. Presence of decoding routines
4. Shellcode presence probability
5. Number of direct string assignments
6. Number of event attachments.
7. Number of suspicious objects names.
8. Number of suspicious strings
9. Number of suspicious tag strings

- **URL & HOST BASED Features:**

This features helps to detect this page is malicious or not. Even though detecting drive-by download pages using URL features is more different than in the case of phishing pages or scam pages, some information enclosed in the URL and coupled with the referenced swarm is used to assist in the detection.

1. Number of suspicious URL patterns.
2. Presence of a subdomain in URL
3. Presence of IP address in URL
4. Value of the TTL for the DNS A record
5. Relationship between PTR and A records
6. Registration date
7. Country Code

IV. RESULT & ANALYSIS

The proposed method performance using three metrics: accuracy in phishing website detection, accuracy in legitimate website detection as well as the page load delay incurred while testing. I compare Proposed method results to six other antiphishing toolbars: IE7 (Internet Explorer 7), Mozilla Firefox 2.0, Mozilla Firefox 3.0, SpoofGuard, Netcraft Toolbar and Google Chrome.

Technology	**Detection Technique**	**Blocked**	**Warned**
Proposed Method	Feature Extraction	Custom pop-up, Overlay blocks user	NA
Firefox 2.0	Blacklist	Custom pop-up page	NA
Chrome	Blacklist	Redirects to help page	NA
IE7	Blacklist & Heuristics	Redirects to help page	Pop-up bar
Netcraft	Blacklist & Heuristics	Pop-up blocks progress	Red bar displayed in toolbar
SpoofGuard	Heuristics	Pop-up blocks progress	Yellow circle

Table 1: Result Analysis

Blocking and Warning Users

Previous research has demonstrated that active security indicators are far more effective than passive security indicators [95]. IE7, SpoofGuard and Netcraft all employ both active and passive security indicators while Proposed method, Firefox 2.0, Firefox 3.0 and Chrome only use active security indicators. If an anti-phishing tool uses an active security indicator i say that the tool has \blocked" the phishing attack. If the anti-phishing tool presents a passive indicator we say that the tool has warned the user about the phishing attack. The indicators presented by each of the tools is summarized in Table 1 as well as the type of detection (blacklist, heuristic,
Information Retrieval employed by the tool). Blocking is so much more effective than warning that I compare tools in terms of blocking in our results section.

I see that WebCrawler is able to extract links from a wide range of content types and especially HTML pages contain many links that we can use for our crawl. We need to consider that only discovered links which do pass the requirements are queued for crawling. These requirements do contain, among others, whether a URL is already queued or downloaded, if a URL is still unique after normalization, and if a URL is in the scope of the crawl.

Module	Handled Objects	Extracted Links	Links / Objects
HTTP Extractor	6,487,350	735,538	0.11
HTML Extractor	4,132,168	345,231,030	83.55
CSS Extractor	19,881	88,307	4.44
JS Extractor	21,658	162,291	7.49
SWF Extractor	17,921	11,117	0.62
URL Extractor	6,506,489	6,776,544	1.04
XML Extractor	35,165	1,638,260	46.59

TABLE 2: LINK EXTRACTION

V. CONCLUSION

In this paper describe a novel approach for a filtering technique to finding malicious web pages very effectively and efficiently using supervised machine learning. It observes existing systems with their problems in various situations and provides detailed study of some other techniques for researcher to research to analysis and detection of malicious web pages. It does summarize that feature extraction technique used in this system affects the overall performance of system. Then projected a comprehensive scheme that addresses all security issues like integrity, privacy, confidentiality simply throughout webpage feature extraction.

From initial results, it is proven that web page feature extractions are much important in malicious web page detection project. Future work of this project will be implementation of advanced feature extraction algorithms which able to give high security to identify malicious web pages. Also As per the result using large scale detection datasets 85% of accuracy shows for analysis and detection of malicious web pages.

VI. ACKNOWLEDGMENT

I take this opportunity to express my sincere gratitude towards the Department of Computer Network Engineering, G.H.Raisoni, Ahmednagar, India that gave me an opportunity for presenting the paper in their esteemed organization. I would like to express my sincere thanks to my PROJECT guide Prof. Amruta Amune for her constant support and experienced guidance and providing me with adequate facilities, precious help and advice without which the successful completion of this seminar would not have been possible. I am also thankful to H. O. D. & Principal of GHRCEM for giving me necessary resources and support to complete my Project Dissertation. Last but not the least, I thank all others, and especially my classmates and my family members who in one way or another helped me in the successful completion of this work.

REFERENCES

[1] Marco Cova, Davide Canali, Giovanni Vigna and Christopher Kruegel, Prophiler: A Fast Filter for the Large Scale Detection of Malicious Web Pages, in proceedings of Internation World Wide Web Conference 2011,ACM 978-1-4503-0632-4/11/03.

[2] N. Provos, P. Mavrommatis, M. A. Rajab and F. Monrose, All Your iFrames Point to Us, in proceedings of USENIX Security Symposium, 2008.

[3] M. Cova, C. Kruegel and G . Vigna, Detection and Analysis of Driveby- Download Attacks and Malicious Javascript Code ,in procedings of International World Wide Web Conference (WWW), 2010

[4] J. John, F. Yu, Y. Xie, M. Abadi and A. Krishnamurthy, Searching the Searchers with SearchAudit, in proceedings of USENIX SecuritySymposium, 2010.

[5] J. Stokes, R. Andersen, C. Seifert and K. Chellapilla, WebCop : Locating Neighborhoods of Malware on the Web , in proceedings of USENIX Workshop on Large-Scale Exploits and Emergent Threats,2010.

[6] Luca Invernizzi , Marco Cova, Paolo Milani Comparetti, Giovanni Vigna and Christopher Kruegel, EVILSEED:A Guided Approach to Finding Malicious Web Pages, IEEE Transcation, vol.25,pp 221-245,2012.

[7] Van Lam Le , Ian Welch , Xiaoying Gao,and Peter Komisarczuk , Identification of Potential Malicious Web Pages ,in proceeding of 9th Australasian Information Security Conference (AISC),vol.116,2011.

[8] N. Provos, D. McNamee, P. Mavrommatis, K. Wang and N. Modadugu, The Ghost in the Browser: Analysis of Web-based Malware. In processing of USENIX Workshop on Hot Topics in Understanding Botnet, 2007.