IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 25-30
www.iosrjournals.org

Malicious URLs Detection and Classification Methodologies

[1]Himani Jangra, [2]Chander Diwaker, [3]Atul Sharma
[1,2,3](Computer Engineering Department, U.I.E.T., Kurukshetra, India)

Abstract: *Malicious URL detection has become increasingly difficult due to the evolution of phishing campaigns and efforts to avoid attenuation black list. The current stateof cybercrimehas allowedpiratesto hostcampaignswith shorterlife cycles, which reduces the effectiveness of theblacklist.Asthe same time, normal supervised learning algorithms are known to generalize in specific patterns observed in the training data, which makes them a better alternative against piracy campaigns. However, the highly dynamic environment of these campaigns requires models updated regularly, which poses new challenges as most typical learning algorithms are too computationally expensive retraining.*

Keywords: *Computer Security, Adware Classification, Malicious web page analysis, Machine Learning*

I. INTRODUCTION

Ad ware, short for Malicious Software advertising is a sequence of instructions that perform malicious activities on a computer network[1]. The history of malware began with "computer virus", a term introduced by Cohen. This is a piece of code that replicates by attaching itself to other executable in the system. Today, the malware includes viruses, worms, Trojans, root kits, backdoors, bots, spyware, adware, scare ware and any other program that has malicious behavior. Adware is a fast growing threat to modern computer networks. Production Adware has become a multi-billion. The growth of the Internet, the advent of social networks and the rapid proliferation of botnets has caused an exponential increase in the amount of Adware. In 2010, there was a sharp increase in the amount of Adware spread through spam emails sent machines that were part of botnets. McAfee Labs reported that there were 6 million new infections each month [2].

Two critical elements affecting mobile use are privacy and positive user experience. The market for mobile applications is based on trust. Mobile advertising is questionable practice, such as applications that use deceptive practices adware, a negative impact on the perception of privacy of the end user and the user experience. Do things like capture personal information such as email addresses, Device ID, IMEI, etc. without properly notifying users and phone settings and change desktop without consent, it's annoying and unacceptable for mobile users. While most mobile ads are not malicious, however, they are undesirable for most people.

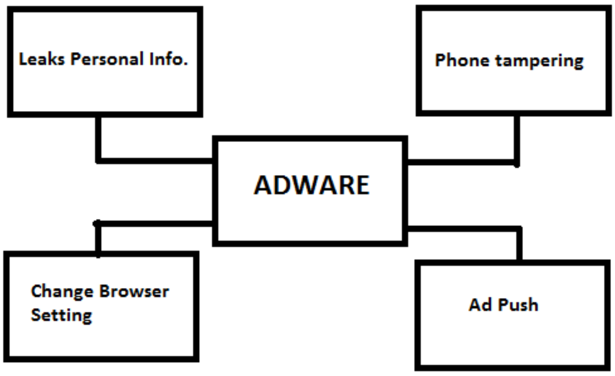

Fig 1: Dangers of Malicious URLS and Adwares

Asmalware ontheInternetspreadsand becomes moresophisticatedanti-malwaretechnologymust be improvedin order toidentify new threatsin an efficient manner, and most importantly, automatically.Malicious Web contenthas become oneof the mosteffective mechanisms forcyber criminalsto distributemalicious code.In particular, attackers often usedrive-by-download exploitsto compromise alarge number of users. To make adrivebydownloadattack, the attacker craftsfirst codemaliciousclient-side script(usuallywritten in JavaScript) that targetvulnerabilitiesin a web browserorin one ofthe browserplugins.This codeis injected intocompromised websitesorsimplyhosted on a serverunder the controlof criminals.When a victimvisits a maliciouswebsite, the malicious code is executed,and ifthe browserof the victimis vulnerable, the browser is compromised.As a result, the computer of the victimis usuallyinfected with malware. Giventhe growingthreat ofmaliciousweb pages, itis not surprisingthat researchers havebegun investigatingtechniques to protectWeb users.Currently, the most widespreadprotectionis based on theURL blacklists. [4] Theseblacklists(such as Google Safe Browsing) store theURLs that havebeenfound to be malicious. The lists areinterviewed by abrowser beforevisitinga web page.When the URLis onthe black list,the connection is brokenorwarning.Of course, tobe able tobuild and maintainsuch ablacklist, automated detection mechanismsare neededthat can findon the internetweb pagecontaining maliciouscontent.

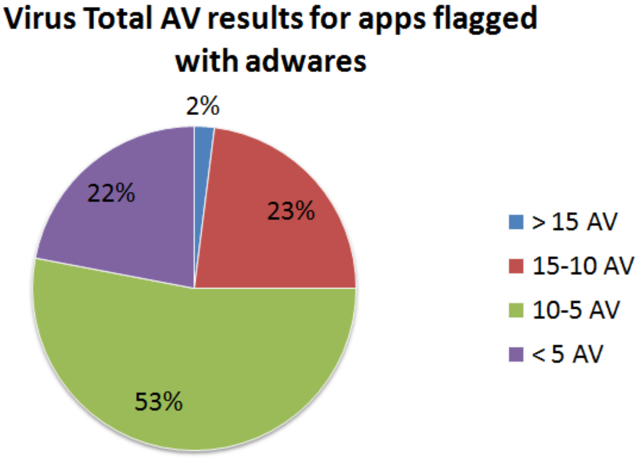

Fig 2: Virus Total results flagged with Adware [5].

InanalogyVirus Total[5]is a free online servicethat analyzesmalicious filesand URLs. It facilitates thequick detection of viruses, worms, Trojans,and all kinds ofmalware.Servicesstoresall analyzescarried out,which allowsusers to search foragivenreportingMD5,SHA1,SHA256orURL.Repliesresearches return thelastanalysison the resourceof interest.Theservicealso allows you tosearch thecommentsusers poston filesand URLs,inspect ourpassiveDNS dataand retrieveinformationdetailsof the threatondomains andIP addresses.Read more aboutresearchwith the Service.Virustotal resultstagged withadwareup to 53%.

Search enginestypicallyindexhugeamountsof information,andas suchcan be considered asvery comprehensiverepositoriesof knowledge.Afterthe heuristicdescribed above, we propose to use the searchresults themselvesto better understandadditionalrequestforinterpretation.Usingrelevancefeedbacknicknameparadigm, andassume thesearch resultsto be relevantto the query.Certainly,notall resultsare as relevant, so hackers canuseelaboratevoting systemsto obtainreliable knowledge aboutthe query totacklecommon user.

II. Related Work

Significant scrutiny has been finished in the globe of computer protection for the detection of understood and unfamiliar malware retaining disparate contraption discovering and data excavating approaches. The authors utilized two static features removed from malware and benign multimedia, Intention Length Frequency (FLF) and Printable Thread Data (PSI) [6]. This work was instituted on the hypothesis that "though intention calls and strings are self-governing of every single solitary supplementary authors underpin every single solitary supplementary in categorizing malware". Disassembly of all the examples was finished retaining IDA Pro and FLF, PSI features were removed retaining Ida2DB.

In the work, intention length is the number of bytes of plan in the function. Frequencies of all intention lengths for all malware was computed and distributed in the exponential interval scopes when finished it came up to be 50 intervals. Printable Thread Data in every single solitary unpacked malware was removed and all the strings for all malware were joined to craft a database. A dataset was crafted alongside these strings as 5 features that accord binary worth of whether a particular malware encompassed this thread or not. All strings alongside minimum length of 3 were selected. With the selected features 13 disparate datasets were crafted for 13 disparate malware families and benign programs. And the authors utilized 5 classifiers; Naive Bayes, SVM, Random Forest, IB1 and Decision Table. Best aftermath was obtained by AdaBoostM1 alongside Decision Table alongside an accuracy rate of 98.86%. It was additionally noted that the aftermath obtained by joining both features were supplementary satisfactory than retaining every single solitary kind of features individually. Schultz *et al.* [7] utilized disparate data excavating methods to notice unfamiliar malware. The authors amassed 4,226 strategies of that 2.365 were malicious and 1,001 were benign.

In the selected data there were 206 benign executables and 38 malicious executables were in PE format. Static features from every single solitary design were removed retaining three approaches; Binary profiling, Strings and Byte sequences. Binary profiling was demanded to merely PE files and supplementary methods were utilized for all programs. Binary profiling was utilized to remove three kinds of features;

1) Catalog of Vibrant Link Libraries utilized by the PE,
2) Intention calls made from every single solitary Vibrant Link Library and
3) Exceptional intention calls in every single solitary DLL. "GNU" design was utilized to remove printable strings. Every single solitary thread was utilized as a feature in the dataset. In the third method for features extraction, hex dump [8] utility was used. Every single solitary byte sequence was utilized as a feature. The authors demanded regulation instituted discovering algorithm RIPPER [9] to the 3 datasets alongside binary profiling features, Naïve Bayes classifier to data alongside Thread and Byte Sequence features and in the conclude six disparate Naïve Bayes classifiers to the data alongside Byte 6 Sequence features. To difference the aftermath from these methods alongside instituted signature instituted method, the authors projected an automatic signature generator. With RIPPER they came to be accuracies of 83.62%, 89.36%, and 89.07% suitably for datasets alongside features DLLs utilized, DLL intention calls and Exceptional Calls in DLLs. The accuracies obtained alongside Naïve Bayes and Multi-Naïve Bayes were 97.11% and 96.88%. The result using Signature method came to be 49.28% accurate. Multi-Naïve Bayes produced larger aftermath contrasted to the supplementary methods.

The data in PE headers was utilized for the detection of malware[11]. The work was instituted on the assumption that there ought to be difference in the characteristics of PE headers for malware and benign multimedia as both were industrialized for disparate purposes.1908 benign and 7863 malicious executables were collected. The malware examples encompassed viruses, email worms, trojans and backdoors. PE headers of all the files were dumped retaining a design yelled DUMPBIN. Every single solitary header (MS DOS header, file header, discretional header and assisting headers) in the PE was trustedas a probable attribute.

For every single solitary malware and benign design locale and entry benefits of every single solitary attribute were calculated. For the reloc assisting in the PE, the decision of whether a malware encompassed that assisting or not was noted down. Every single solitary earth in the dataset was adjusted to binary worth in the attribute linearization process. Unimportant and redundant qualities were eliminated in the consecutive step. Unimportant qualities are the ones that were present in merely one executable. Redundant qualities were the ones present in all executables.

In parallel, attribute selection was provided retaining Prop Vector Machines. The growing dataset was tested alongside SVM classifier retaining five-fold cross validation. Accuracies of 98.19%, 93.96%, 84.11% and 89.54% wereobtained for virus, email worm, trojans and backdoors respectively. The detection rates of viruses and email worms were elevated contrasted to the detection rates of Trojans and backdoors. In Kolter*et al.* [10], countless byte sequences from the executables were used. T he authors amassed 1971 clean and 1651 malicious executables. All of them were in PE format. Hexadecimal plan for every single solitary executable was obtained by retaining hex dump [10]. From that plan countless bytes in sequence were joined to produce n-grams. Training data was synchronized alongside the removed n-grams as binary features. Most relevant features were selected by computing the data gain for every single solitary feature. As a result 500 features were selected. Countless data excavating methods like IBk, TFIDF, naive Bayes, Prop Vector Mechanisms (SVM) and decision trees demanded to produce regulations for categorizing malware. The authors additionally utilized "Boosted" naïve bayes, SVM and decision tree learners. Three examinations were managed on the data. In the main examination, size of the words, size of n-grams and number of features appropriate for the examinations were assessed. From the subset of executables, n-grams were removed alongside n=4. Countless data excavating examinations were managed to find the optimal size of n-grams by fluctuating the subset size (10, 20, 100, 1000 etc).

Best aftermath was obtained alongside the size of 500. By fixing the size to 500, n in n-grams was varied and the aftermath were supplementary precise alongside n=4. In the consecutive examination, out of 68,774,909 n-grams, 500 best n-grams were selected and demanded 10-fold cross validation in every single solitary association method. In the third examination, 255 million n-grams were removed from the all the executables and t the 8 comparable procedures were pursued as in consecutive experiment. The boosted classifiers, SVM and IBk produced good aftermath contrasted to the supplementary methods. The presentation of classifiers was enhanced by boosting and the finished presentation of all the classifiers was larger alongside the large dataset contrasted alongside the puny dataset.

Dmitry and Igor [11], utilized positionally reliant features in the Original Entry Point (OEP) of a file for noticing unfamiliar malware. In the work, authors utilized 5854 malicious and 1656 benign executable in WIN 32 PE format. Varied data excavating algorithms like Decision Table, C4.5, Random Forest, and Naive Bayes were demanded on the synchronized dataset. Three assumptions were made for the work.

1) Studying the entry point of the design understood as Original Entry Point (OEP) reveals supplementary precise information.

2) The locale of the byte worth of OEP address was set to zero. And the offsets for all the bytes in OEP was trusted to be in the scope

3) Merely a solitary byte can be elucidating for every single solitary locale value. So the scope for Byte in locale worth is from 0 to 255. And in the conclude the probable number of features that could be utilized for association was 65536. The dataset encompassed three features; Feature ID, Locale and Byte in Position. Feature selection was provided to remove supplementary momentous features. The features removed in this pace were instituted on the dependencies amid features data gain and the center constituents of the features. The growing data was tested opposite all classifiers and the consequences were contrasted instituted on ROC-area.

Random Forest outperformed all the supplementary classifiers. A Specification speech was derived in Jha*et al.* [12] ,instituted on the arrangement calls made by the malware. The specifications were hypothetical to delineate the deeds of 9 malware. The authors additionally industrialized an algorithm understood as MINIMAL that mines the specifications of malicious deeds from the dependency graphs and demanded this algorithm to the email worm Bagle.J, a variant of Bagle malware.

Clean and malicious files were provided in the manipulated nature, traces of arrangement calls were removed for every single solitary example as execution. Dependency graph was crafted retaining arrangement calls and the argument dependencies. In the graph, every single solitary node denotes a arrangement call and its arguments; every single solitary frontier denotes dependency amid arguments of the two system calls. A sub graph was removed from the malware dependence graph by contrasting alongside benign multimedia dependence graph such that it exceptionally specifies the malware behavior.

A new file alongside these specifications ought to be categorized as malware. Virus prevention Flawless (VPM) to notice unfamiliar malware retaining DLLs was demanded by Wang *et al.*[13] in the work. 846 malicious and 1,758 benign files in handy executable format were collected. All files were parsed by a design “dependency walker” that displays all the DLLs utilized in a tree structure. Three kinds of qualities T1, T2 and T3 were derived from the growing tree. T1 is the catalog of APIs utilized by main design undeviatingly, T2 indicates the DLLs implored by supplementary DLLs supplementary than main design and T3 is the connections amidDLLs that consistsof dependency tracks down the tree. At the end, 93,116 qualities were obtained. The qualities alongside low Data Gain were removed. Extra feature reduction was finished by retaining L-SVM. As a result, 10 429 vital qualities were selected and tested the dataset alongside RBF-SVM classifier retaining five-fold cross validation. The detection rate alongside RBF-SVM classifier was 99.00% alongside Real Affirmative rate of 98.35% and Fake Affirmative rate of 0.68%. A similarity compute method for the detection of malware was counseled by Chanted *et al.* [14] instituted on the hypothesis that, variants of a malware have the comparable core signature that is a combination of features of the variants of malware. To produce variants for disparate strains of malware, instituted obfuscation methods were used. Generated variants were tested opposite 8 disparate antivirus products. Four virus strains W32.Mydoom, W32.Blaster, W32.Beagle and Win32.Wika were utilized in this process.

The new malware strains obtained from obfuscation were categorized into 5 types; null procedure and dead plan insertion, data modification, manipulation flow modification, data and manipulation flow modification, and pointer aliasing. The basis plan of every single solitary PE was parsed to produce API yellingsequence and the sequence was trusted as signature for that file. Every single solitary API call was given an integer ID. The sequence of API calls was embodied by corresponding sequence of IDs. The consequence in sequence was contrasted alongside the main malware sequence to produce similarity measure. The similarity measures were computing retaining Euclidian Distance, sequence alignment and disparate similarity intentions encompassing cosine compute, range Jacquard compute and Pearson correlation measure.

A mean worth of all the measures was computed for every single solitary signature. The biggest index in the similarity table denotes to that main malware the particular variant belongs. By contrasting that worth alongside a threshold the nature of the file, benign or malicious was decided. The detection rate of SAVE was significantly larger than antivirus scanners. A strain of Nugacheworm was reversed in order to notice its underlying design, deeds and to comprehend attacker’s method for discovering vulnerabilities in a system [15]. In supplement to that, the authors additionally reverse engineered 49 malware executables in a remote nature, removed varied features like MD5 hash, printable strings, number of API calls made, DLLs accessed and URL referenced. Retaining these features they synchronized a dataset. Due to the multi dimensional nature of the dataset, a contraption discovering instrument, BLEM2 [16] instituted on rough set theory was utilized to produce vibrant outlines that ought to assistance in categorizing an unfamiliar malware.

As the size of the dataset was puny, a tremendously insufficient number of decision regulations were generated and the aftermaths were not satisfactory. Instituted on vibrant scrutiny [17], spatial -temporal data in API calls was utilized to notice unfamiliar malware. The counseled method consists of two modules; an offline module that develops a training flawless retaining obtainable data and an online module that generates aassessing set by removing spatial-temporal data across the training flawless to categorize run era procedure as whichever benign or malicious. Arrangement logs for 100 benign and 416 malicious strategies were amassed and 237 innate Windows API calls of disparate clusters like socket, recollection association, threads etc were sketched and utilized as base. In the vibrant scrutiny, spatial data was obtained from intention call arguments, revisit benefits and were rip into seven subsets socket, recollection 12 association, procedures and threads, file, DLLs, registry and web association instituted on the functionality.

Temporal data was obtained from the sequence of calls and the authors noted that a slight of the sequences were present merely in malwares and were missing in benign programs. Spatial data was quantified retaining statistic and data theoretic measures. By computing the autocorrelation the authors were able to remove the relation amid calls in API call sequences. The authors found no correlation at all and 1 denotes immaculate correlation for that the lag worth ought to be Zero. For the API call sequences best correlation was obtained at n=3, 6, 9... API call sequence was modeled retaining discrete era Markove shackle that enables them to select how countless lags to scrutinize in API sequences and to cut the size of the example space.

The Markov shackle had k states and the transition probabilities amid these states were embodied in state transition matrix T. Every single solitary transition probability was trusted as a probable attribute. Feature selection was provided to select qualities alongside most data gain. In the conclusion they selected 500 transitions and synchronized a set alongside Boolean values. Three datasets were crafted by joining benign strategies API sketch alongside every single solitary malware type. The three datasets were combinations of benign-Trojan, benign-virus and benign-worm.

Authors managed two experiments. Main one was to notice the joined presentation of spatio-temporal features contrasted to standalone spatial or temporal features. Consecutive examination was managed to remove a negligible subset of APIclusters 13 that gives comparable accuracy as from the main experiment. For this, the authors joined API call clusters in all probable methods to find the negligible subset of clusters that ought to give comparable association rate as obtained in main experiment. From the main examination, the authors obtained 98% accuracy alongside naive bayes and 94.7% accuracy alongside J48 decision tree and they came to be larger aftermath alongside joined features contrasted standalone features. The detection rate of Trojans was less contrasted to viruses and worms. In the consecutive examination, combination of API calls related to recollection association and file I/O produced best aftermath alongside an accuracy of 96.6%.

The work was instituted on an assumption that, deeds of a malware can be exposed totally by providing it and discerning its aftermath on the working nature.For this task, the system which was developed grabbed encompassing registryattention,files arrangement attention, web attention, API Calls made, DLLs accessed for every single solitary executable by running them in a remote environment. Retaining the removed features from the reverse engineering procedure, we synchronized three datasets. To these datasets, we demanded data excavating algorithms C4.5, Naïve Bays and a rough set instituted instrument BLEM2 to produce association regulations and contrasted the results. In a slight of the above remarked works [16] merelystatic features like byte sequences, printable strings and API call sequences were used. Nevertheless competent in noticing malware, they ought to be ineffective if the attackers use obfuscation methods to contain malware. To ascertain this setback, a slight supplementary works.

III. CONCLUSION

URL classification is an important information retrieval task. Accurate classification of search queries benefits a number of higher-level tasks such as Web search and ad matching. As search queries are usually short, they usually carry insufficient information for adequate classification accuracy. To address this problem, the research proposed a methodology for using search results as a source of external knowledge.

In future, for the purpose of the study in malicious URL detection and classification a query will be dispatched to a general web search engine, and collect a number of the highest-scoring URLs. The system crawl the Web pages pointed by these URLs, and classify these pages into Benign or malign.

REFERENCES

[1]. JitendraApteand Marina Lima Roesler. "Interactive Multimedia Advertising and Electronic Commerce on a Hypertext Network." U.S. Patent No. 7,225,142. 29 May 2007.

[2]. Ravula and Ravindar Reddy. “Classification of Malware using Reverse Engineering and Data Mining Techniques.” M.S. Dissertation, University of Akron, CS Dept., 2011.

[3]. Anup K. Ghosh and Tara M. Swaminatha. "Software Security and Privacy Risks in Mobile E-Commerce." In Communications of the ACM,vol.44, issue 2, 2001.

[4]. Justin Ma, Lawrence K. Saul, Stefan Savage and Geoffrey M. Voelker. "Beyond Blacklists: Learning to Detect Malicious Web Sites From Suspicious URLs." in Proceedings of the 15th ACM international conference on Knowledge discovery and data mining, 2009.

[5]. "Gap between Google Play and AV vendors on adware classification", HispasecSistemas, S. L. "Virustotal malware intelligence service." 2011.

[6]. Ronghua Tian, Lynn Margaret Batten, and S. C. Versteeg. "Function Length as a Tool for Malware Classification." in IEEE 3rd International Conference on Malicious and Unwanted Software, pp. 69-76, 2008.

[7]. Yanfang Ye, Dingding Wang, Tao Li, Dongyi Ye and Qingshan Jiang. "An Intelligent Pe-Malware Detection System based on Association Mining." Journal in computer virology, Springer,vol. 4, issue 4, pp. 323-334, 2008.

[8]. William W. Cohen. “Learning Trees and Rules with Set-Valued Features.” In proceeding13th International Conference on Artificial Intelligence, ACM, vol. 1, pp. 709-716, 1996.

[9]. Tzu-Yen Wang, Chin-Hsiung Wu, Chu-Cheng Hsieh, "Detecting Unknown Malicious Executables Using Portable Executable Headers," in IEEEFifth International Joint Conference on INC, IMS and IDC, NCM, pp.278-284, 2009.

[10]. Jeremy A. Kolter and Mercus A. Maloof, “Learning to detect malicious executables in the wild,” in Proc. of the tenth ACM International Conference on Knowledge discovery and Data Mining, pp. 470–478, 2004.

[11]. DmitriyKomashinskiy, Igor Kotenko. “Malware Detection by Data Mining Techniques Based on Positionally Dependent Features.” In Proceedings of IEEE 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, pp. 617-623, 2010.

[12]. M. Christodorescu, S. Jha, and C Kruegel. “Mining specifications of malicious behavior.” in Proc. of 6th Joint Meeting of the European Software Engineering Conference and the ACM Sigsoft Symposium on Foundations of Software Engineering, pp. 5–14, 2007.

[13]. TzuYen Wang, Chin-Hsiung Wu, and Chu-Cheng Hsieh. “A Virus Prevention Model Based on Static Analysis and Data Mining Methods.” in Proceedings of IEEE 8th International Conference on Computer and Information Technology Workshops, pp.288-293, 2008.

[14]. A. Sung, J. Xu, P. Chavez, and S. Mukkamala, “Static Analyzer of Vicious Executables (SAVE).” in Proc. of 20th Annual Computer Security ApplicaitonConference, pp. 326– 334, 2004.

[15]. Burji, S., Liszka, K. J., and Chan, C.-C.,”Malware Analysis Using Reverse Engineering and Data Mining Tools.” In IEEE International Conference on System Science and Engineering, pp. 619-624, 2010.

[16]. Chien-Chung Chan and Sengottiyan S. "Blem2: Leaming Bayes' Rules from Examples using Rough Sets," in IEEE 22nd International Conference of the North American Fuzzy Information Processing Society, pp. 187-190, 2003.

[17]. Faraz Ahmed, HaiderHameed, M. ZubairShafiq, and MuddassarFarooq. “Using spatio-temporal information in API calls with machine learning algorithms for malware detection.” in Proceedings of the 2nd ACM Workshop on Security and Artificial intelligence, pages 55–62, 2009.