Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 5, May 2013, pg.1 – 5

RESEARCH ARTICLE

DETECTION OF MALICIOUS URL REDIRECTION AND DISTRIBUTION

T. Manjula Devi[1], R. Krishnaveni[2]
[1]School of Computing Sciences, Hindustan University, India
[2]School of Computing Sciences, Hindustan University, India

[1] *manjudevi28@gmail.com;* [2] *rskichu10@gmail.com*

Abstract— Web-based malicious software (malware) has been increasing over the Internet .It poses threats to computer users through web sites. Computers are infected with Web-based malware by drive-by-download attacks. Drive-by-download attacks force users to download and install the Web-based malware without being aware of it .these attacks evade detection by using automatic redirections to various websites. It is difficult to detect these attacks because each redirection uses the obfuscation technique. This paper analyzes the HTTP communication data of drive-by-download attacks. The results show significant features of the malicious redirections that are used effectively when we detect malware.

Key Terms: - Web-based malware; drive-by-download attacks; packet capturing

I. INTRODUCTION

Damage resulting from Web-based malware has been increasing. Web-based malware uses a *drive-by-download* technique as its attack methods. Drive-by-download attacks force computer users to download and install malware without being aware of it by exploiting the vulnerabilities in a Web browser or some external components [1]. Figure 1 illustrates a typical drive-by- download attack. A user accessing an *entrance site* is redirected to malicious Web sites in sequence. These consist of three separate Web sites. A *zombie site* redirects the user to the next zombie site or an attack site. The zombie site is used as a stepping stone. An *attack site* exploits the vulnerabilities of the user's Web browser and forces the user to download malware from the *malware distribution site*, which contains malicious script codes or contents. These script codes are difficult to analyze because they are often obfuscated. Therefore, it is not easy to detect zombie-site URLs, attack-site URLs, and malware-distribution-site URLs used in drive-by- download attacks. There are two problems related to drive-by-download attacks. The first problem is that malicious Web sites attack users only when they access the malicious Web sites. This makes it difficult to detect the malicious Web sites because users only access them occasionally. The second problem is that a normal Web site may be compromised, causing it to play the role of an *entrance site* or a *zombie site* in drive-by-download attacks. An infected popular site like a social network service will impact a large number of users. The drive-by-download attack increases the risk to Internet users. There have been numerous research projects regarding drive-by-download attacks based on the measurement and analysis of malicious contents.

Egele et al. [2] illustrated and analyzed malicious JavaScript codes. They proposed building defensive mechanisms into a Web browser to mitigate the threats that arise from drive-by- download attacks. Their study showed a successful approach for mitigating drive-by-download attacks based on malicious script codes.

In this paper, we propose a new detection method for drive-by-download attacks using features of the malicious redirections. Our new method is not limited to JavaScript analysis because we observe a sequence of packets between a Web browser and servers.

Cova et al. [3] presented a method for the detection and analysis of malicious JavaScript codes. They developed a system that uses numerous features and applied machine-learning techniques to discriminate the characteristics of normal JavaScript code. Their system can identify anomalous JavaScript codes by emulating the behaviors and comparing them to the normal JavaScript profile. Their system specializes in JavaScript codes.

In addition to JavaScript codes, this paper covers other features such as HTTP methods and URL information. It should be noted here that the evaluation of JavaScript codes is very expensive. Therefore, although we consider only the existence of JavaScript codes, our method does not analyze the code in detail.

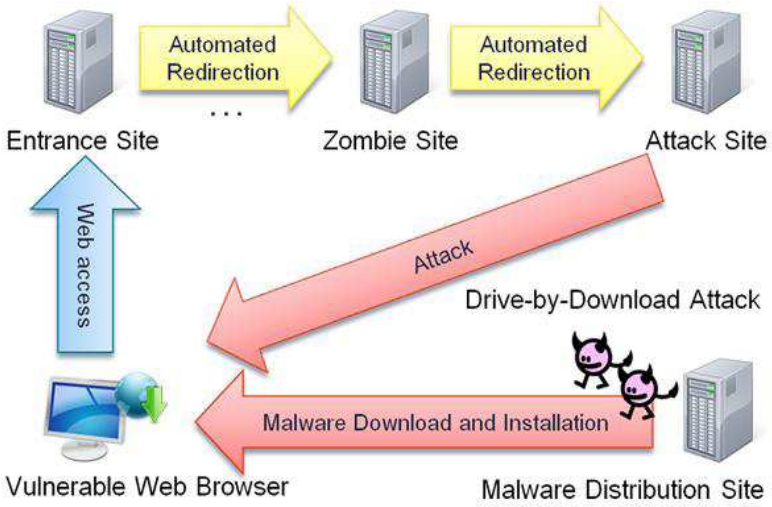
II. EXISTING SYSTEM

Drive-by-download attacks force users to download and install the Web-based malware without being aware of it. A user accessing an entrance site is redirected to malicious Web Sites in sequence. These consist of three separate Websites. Zombie site redirects the user to the next zombie site or an attack site. The zombies it is used as a stepping to one. An attack site exploits the vulnerabilities of the user's Web browser and forces the user to download malware from the malware distribution site, which contains malicious script codes or content. These script codes are difficult to analyze because they are often obfuscated. Therefore, it is not easy to detect zombie-site URLs, attack-site URLs, and malware- distribution-site URLs used in drive-by- download attacks.

III. PROPOSED SYSTEM

A new method for finding the hidden malicious URLs in drive-by-download attacks by analyzing redirections from captured HTTP communication data packets. It is relatively easy to trace redirections in HTTP communication by looking for the referrer fields in GET requests and HTTP responses. However, JavaScript codes can hide a referrer field in the malicious redirections of drive-by-download attacks. The HTTP communication data captured in controlled environment where only the drive-by-download attacks exist. We describe this data-capturing environment later. We try to reveal the features of the redirection to make it possible to detect unknown malicious attacks effectively.

SYSTEM ARCHITECTURE



IV. ALGORITHMS

SIFT ALGORITHM

In this project, used SIFT algorithm on each Web page to identify local visual objects and obtain a global summary. SIFT computes feature descriptors at particular points of interest in an image; these descriptors are invariant to scale, orientation, and affine distortion (and robust to certain types of noise). These properties make SIFT ideal for matching images based on the similarity of their local visual content. The SIFT algorithm detects key points from high contrast regions of the image such as local object edges. Gradient directions and magnitudes are then pkey point. Each key point feature descriptor has 132 dimensions: 128 dimensions for these

orientations, and 4 for its location, scale and rotation. For Web page classification which includes visual features that capture basic statistics of the SIFT key point descriptors. In particular we compute the number of key points as well as the mean and variance of the SIFT key point orientations. These operations yield 257 visual features, which refer to as sift-stats. After using SIFT to identify the local visual objects in each webpage, it also match the local images found in the Web page against a repository of common Web logos. The repository contains the logos of popular banks, e-commerce brands, and social-networking sites. The SIFT matching algorithm recognizes logos in Web pages by identifying the nearest neighbors in its repository; more precisely, it computes the nearest neighbors in Euclidean distance between logo and image key point descriptors. The images in the repository are then assigned a score based on their distance to each local object in the Web page, which include visual features that indicate whether a particular logo was identified in the Web page and, if so, the associated matching score. When an exact match is not found, there may still be a number of candidate matches with roughly the same score. Hence, Incorporate information for the logos which are candidate matches but do not have the highest matching score. It refer to this set of visual features as sift-matching.

OBFUSCATION CHECK ALGORITHM

Using an obfuscation check algorithm, a malicious website determined to be such, depending on the intensity of obfuscation. check the obfu scated JavaScript, proposal system check several point of target URL component. First, the system will check density of target web page. The density elements are the longest unique character stream size and used special character set number. If the longest character stream is over 200 characters, the system can scored 12 point. And give 2 point to target web page URL, if used special character set number is over threshold value. Second, the system will count some feature of target web page for checking frequency score. The system check a particular function's frequency and encoding mark and % symbol occurrence. Finally, web page entropy will be checked for verified obfuscated JavaScript. Whole web code's entropy, total JavaScript entropy, each JavaScript block entropy, variable's name entropy and function's name entropy are checked. Also, the proposal system check other component of web code of target URL, but above descriptions are most important check point in the system. The proposed system checks the website with a hidden I Frame, an obfuscated website, and PE style website. If the website that needs to be checked belongs to any of these types of websites, it will be determined to be a malicious website. If it is determined to be a malicious website, it is sent to an automatic website module in the form of a high interaction client honey pot.

GENETIC ALGORITHM

Genetic Algorithm is to improve the classification speed and precision. A small dataset is collected and expanded through GA mutations to learn the system over short time and with low memory usage. A completely independent testing dataset is automatically gathered and verified using different trusted web sources. They algorithm achieves an average precision of 87%. To generate a larger dataset from the initial dataset we use Genetic Algorithm. GA is a biologically inspired algorithm that applies the principles of evolution and natural selection. The algorithms starts with an initial population encoded as a chromosome structure which is composed of genes encoded as numbers or characters. In our case the initial population represents the group of features for the training dataset. Chromosome goodness is evaluated using a fitness function that uses mutations and crossovers to simulate the mutation of species. The fittest precision attained by adding the new individual features to MALURLs. The features added include TF-IDF, JS- Enable-Disable and 3-4-5 grams. The addition of TF-IDF results in a significant increase in classification precision from 66% to 76% as shown in Table III. This is expected because of the volume of information presented by this feature. Adding the JS-Enable-Disable did show a very good increase in average precision from 63% to 69% as illustrated by Table IV. The experiments to measure the improvement in MALURLs precision achieved by adding3-4-5 grams show a small increase in average precision from 77% to 80% as illustrated by Table V. 3-4-5 grams calculation is complex, takes a long time and puts the user at risk due to the need to download the document. Therefore n-grams can be deemed irrelevant because of the high overhead and small improvement which is consistent with our goal of keeping the algorithm.

V. MODULES

MAC/IP ADDRESSING MONITORING

It defines a session as a chain of packet flows. It can trace a session by sorting with request and response in TCP using the same MAC address, IP address, and port number. A server is identified by an IP address and

domain name1. When we access a Web site, we connect to a Web server or servers, and the Web browser establishes sessions with the Web server or servers.

TIME STAMP

Web browsers have a progressive rendering function that accelerates the rendering of a Web page. This function evaluates data such as HTML files and JavaScript files immediately after the download. It can start by analyzing the data from GET requests and define the test range. HTTP responses are analyzed for 120 seconds time intervals. Time stamp defined this time interval empirically using a preliminary experiment that measured the time interval from the generation of redirection to the download of the first malware. Therefore, this range depends on the network environment, including the bandwidth and number of connected users.

REFFERRER TEST AND URL TEST

The referrer test detects redirections based on the referrer field of the GET request. If the previous URL is set in the referrer field of the GET request, it can trace the redirection. When it is moves to a newly clicked URL, the origin site URL is set as the referrer field. The URL test detects redirections by picking up URLs from an HTML content file and a location field from an HTTP response header. Thus, the URL test covers two methods. The first method analyzes all of the characters in an HTML content file and extracts URLs. The second method gets a URL from the location field in an HTTP response header.

HOST VERIFICATION –CONTENT DESCRIPTION

In Host Verification, it collects the list of all of the servers and sessions in the communication data then call this list the first list. Next, it analyzes the communication data again and performs the URL test and referrer test for each entrance Web site. Some servers and sessions are found to be redirected. By content description, apply the redirected servers and sessions into the other list. When call the list of second list, Finally if any sessions in the first list are not covered by either the URL test or referrer test it investigate the remaining HTTP communications in detail. If there are any sessions in the first list with the known servers that appear in the second list these sessions are classified as redirected sessions and belong to the same redirection group as the URL test and referrer test.

VI. CONCLUSION

It analyzed communication data captured in an environment where only the communication data of drive-by-download attacks existed. It found the significant features of malicious redirection. The new methods successfully detected the redirections by using these features. Our future plans are to evaluate the extraction method of malicious redirections by using the acquired features from normal and malicious communication data and to apply the proposed methods to communication data captured in various networks.

VII. FUTURE ENHANCEMENT

This project concludes that by using HTTP Communication data and URL direction and distribution is achieved. It found the significant features of malicious redirection. It verifies whether the website is a malicious website or not and it prevents drive by download attacks on the web browser. The existing systems have different techniques and algorithms to avoid the drive by download attacks. It is performed by feature based URL redirection.

REFERENCES

[1] Moshchuk, A.; Bragin, T.; Gribble, S.D.; Levy, H.M. A crawler-based studyof spyware on the Web. University of Washington, 2006; Vol. 2.
[2] Egele, M.; Kirda, E.; Kruegel, C. Mitigating drive-by download attacks: challenges and Open Problems. INETSEC 2009 IFIP AICT 309; Vol. 4, pp. 52-62.
[3] Cova, M.; Kruegel, C.; Vigna, G. Detection and analysis of drive-by-download attacks and malicious JavaScript code. WWW 2010; Vol. 4, pp. 281-290.
[4] Hatada, M.; Nakatsuru, Y.; Akiyama, M.; Miwa, S. Datasets for anti-malware research ~MWS 2010 datasets~. Computer Security Symposium 2010; Vol. 10, pp. 19-21.
[5] Akiyama, M.; Iwamura, M.; Kawakoya, Y.; Aoki, K.; Itoh, M. Design and implementation of high interaction client honeypot for drive-by-download attacks. IEICE Transactions on Communications,

2010; vol. 5, pp. 1131-1139.

[6] Takata, Y.; Mori, T.; Goto, S. Redirect analysis of web-based malware. The 73rd National Convention of IPSJ, 2011; Vol. 3.

[7] Monther Aldwairi ; Rami alsalman. MALURLS: A light weight malicious website classification based on URL features. Emerging technologies in web intelligence,vol 4,No 2,May 2012

