International Journal of Communication Networks and Security

Volume 1 | Issue 2 Article 12

2011

Integrated Approach of Malicious Website Detection

Krishnaveni Raju
Department of Computer Science and Engg. Anna University , Chennai- 60025 TamilNadu ,India., rskichu10@gmail.com

C. Chellapan Dr.
Anna University , Chennai- 60025, TamilNadu ,India, drcc@annauniv.edu

Integrated Approach of Malicious Website Detection

Krishnaveni Raju[1], C.Chellappan[2]
Department of Computer Science and Engg.
Anna University , Chennai- 60025 TamilNadu ,India.
rskichu10@gmail.com[1], drcc@annauniv.edu[2]

BSTRACT : With the advent and the rising popularity of Internet, security is becoming one of the focal point. At present, Web sites have become the attacker's main target. The attackers uses the strategy of embedding the HTML tags, the script tag to include Web-based Trojan scripting or redirector scripting, the embedded object tag which activates the third-party applications to display the embedded object and the advanced strategy is the ARP spoofing method to build malicious website when the attackers cannot gain control of the target website. The attacker hijacks the traffic, then injects the malicious code into the HTML responses to achieve virtual malicious websites. The malicious code embedded in the web pages by the attackers; change the display mode of the corresponding HTML tags and the respective effects invisible to the browser users. The display feature setting of embedded malicious code is detected by the abnormal visibility recognition technique which increases efficiency and reduces maintenance cost. Inclusion of the honey client increases the malicious website detection rate and speed. Most of the malicious Web pages are hence detected efficiently and the malicious code in the source code is located accurately. It can also handle End-User requests to know whether their webpage is free of Malicious codes or not.

KEYWORDS
Abnormal visibility, Attacker's mechanism, Malicious Websites, Detection mechanism, Honey client, JavaScript interpreting.

I. INTRODUCTION

With the increased usage of Web application, Websites have become the attackers' main target. Nearly one third of the Web sites contain malicious code. Malicious codes are pieces of code that can affect the secrecy, the integrity, the data, control flow and the functionality of a system. Therefore, their detection is a major concern within the computer science community as well as within the user community.

Web sites containing the malicious code generated by JavaScript has accounted in large numbers. When the users access the Web pages containing malicious code with any Web browser, some mal operations are performed by the system. The proposed system can interpret and execute scripts efficiently and detects the malicious codes generated by scripts, especially the encoded scripts.

The attackers while embedding Trojan horses in the Web pages can change the display mode of the corresponding tags in the Web pages. Therefore, some feature was added to the existing system to describe the display feature detection of malicious web pages.

This system uses Web spider and HTML parser to fetch and parse the Web pages from target site. The system parse the codes of the Web pages and convert them into the data structures which is easily recognized by the detection engine, further matching these structures with the abnormal visibility fingerprints and locating the malicious codes in the source codes of pages accurately and efficiently. The maintenance of these finger prints is very low. The finger print library can be updated by defining new ones for further development.

The rest of this paper is organized as follows. Section 2 describes related work. Section 3 gives the mechanisms in malicious websites. Section 4 describes the malicious website detection methods. Section 5 gives the experimental results, section 6 concludes this paper and section 7 shows the references.

RELATED WORKS

VeriWeb detects the executable paths of Web sites which can be exploited by attackers. It can analyze JavaScript scripts using string matching. However, it can hardly match all sort of scripts used to embed malicious codes, especially encoded scripts. [M. Benedikt et al., 2002]. Intrusion-detection is performed by attacking behavior pattern matching, which mines the server logging in real time [M. Almgren et al., 2000]. New signatures have to be built for different sites. Honey Monkeys is costly and slow [Y. Wang et al., 2006] and HoneyC is also based on signatures of malicious codes. [C. Seifert et al., 2007] Both are based on Honey Pot technique. The crawler based system in [A. Moshchuk et al., 2006] and Monkey- Spider [A. Ikinci., 2008] download possible malicious pages to local machines scan them with anti-virus software's. However special conditions cannot be resolved and timely updations cannot be obtained. StopBadWare.org [http://www.StopBadWare.org.] is an organization established to detect malicious Web sites. A product called Sucop also detects malicious Web pages [http://www.sucop.com/]. Various anti-virus software's can also detect the malicious codes passively at browser-end. HTML Parser is a tool which parses the malicious codes of the web pages [http://htmlparser.sourceforge.net/].

All the above works are based on the virus signature, resulting in the large quantity of maintenance and the disadvantages of update. The methods based on the virus signature can't be with high performance because of string matching.

MECHANISMS IN MALICIOUS WEBSITES

The victims who visit the malicious websites will be redirected and exploited by the Web-based Trojans, implemented in scripting languages including JavaScript.

Strategies for Redirecting Visitors to Web-based Trojans

To redirect the visitors to the actual Web-based Trojan, attackers are typically using one of the following three categories of strategies.

Embedded HTML Tags : Embedded HTML tags such as iframe, frame, and others, are used to embed the Web-based Trojan into the source code of the website. The aim of this HTML element is to create an inline frame that displays another document. When that page is opened, the included document is displayed in the inline frame. Thus attackers take advantage of this to include the Web-based Trojan directly, setting the iframe to be invisible by setting the height or width of the iframe to zero or a very small value, for example:

```
<iframe src="URL to Trojan" width="0" height="0"
frameborder="0"></iframe>
```

Trojan horse : Trojan horse attacks are accomplished by inserting malicious code into other people's programs. When the user executes their program, they unintentionally execute the Trojan horse program. Trojan horse programs may be used by criminals to commit fraud, embezzlement, sabotage and espionage. Many current web sites insert a small piece of code like a cookie into your browser file, which may contain a Trojan horse.

Malicious Scripts : The second category uses the script tag to include Web-based Trojan scripting or redirector scripting, which are often XSS (Cross-Site Scripting) vulnerabilities.

Embedded Objects : The third category of strategies is based on the embedded object tag for activating third-party applications like Flash or Baofeng media player or Browser Helper Objects to display the embedded object. When vulnerabilities in these applications and BHOs are found, attackers then use this strategy to inject the objects to the vulnerable applications, which exploit them in order to remotely execute code on the victim's machine.

ARP Spoofing : This is another advanced strategy to build malicious website when the attackers cannot gain control of the target website. The attacker uses ARP spoofing in order to act as a Man-in-the-Middle, and hijacks all of the traffic from and to the victims in the same Ethernet subnet. The attacker then injects malicious code into the HTML responses from the target website, or all of the web traffic, to achieve virtual malicious websites.

DETECTION METHODS

Malicious code hidden in the websites is detected by the following ways:

- Reactive detection
- Automatic detection
- Proactive detection
- Infrastructure based detection
- Abnormal visibility recognition detection.

Integrated approach

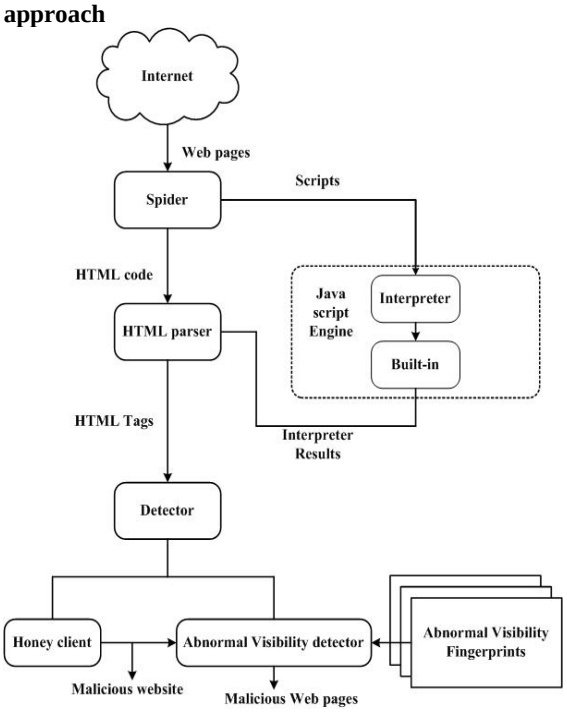

Fig. 1. Malicious website detection architecture

An integrated approach of using honey client with the abnormal visibility recognition detection is chosen so as to increase the malicious website detection rate and speed along with the increased performance and efficiency.

Spider : It automatically detects the malicious Web pages in Web sites. It starts crawling from an initial URL, fetches the HTML source code of the page and extracts the related link URLs. The contents are then fed to HTML parser.

HTML Parser : During the extraction and detection process, the tags in HTML pages should be represented in a well-formed structure format. A Java HTML parsing library is used for implementation. A class by name IFrameTag is implemented for parsing iframe tags. The parser converts the HTML codes to the linked list structure of tags.

JavaScript Engine : It consists of a Rhino-based JavaScript interpreter and some necessary simulation built-in objects. The scripts extracted from the original Web page is fed to the JavaScript engine which interprets and execute them. The output data received by the built-in objects is integrated with the HTML codes of original Web page and then fed to the parser.

Honey client : Client honeypots are devices for detecting malicious servers on a network. It generates a queue of server requests, issues these requests to the servers one-by-one and gets the response of the servers. After a response is obtained, the client honeypot perform an analysis that determines whether the server is malicious or benign. In this system, by using the client honeypot, detection of malicious websites are increased within a short time duration.

Abnormal visibility detector : It is used to detect abnormal visibility. The abnormal tag is determined by

matching its attributes values with the abnormal visibility fingerprints. For HTML codes, the detector will check the width and height attributes of related HTML tags directly. For JavaScript scripts, detection will be performed to the tags after interpretation and execution. The width and height values are compared with a threshold. If they are less than the threshold, it is considered that there is an abnormal visibility in the Web page and suspected as a malicious webpage. If the display style of the iframe tags are set to "display: none", the width and height values are regarded as zero. The detection will be carried out based on some abnormal visibility fingerprints such as

- Abnormal width or height.
- Abnormal display: none display style.
- Abnormal iframe generated by scripts.

Sometimes websites may introduce invisible iframe tags for normal application functions. To avoid false positive in such cases, the detector considers only the invisible iframe tags whose srcs link to external sites.

EXPERIMENTAL RESULTS

Using the system developed, various abnormal visibilities are distributed as shown in the Fig.2.

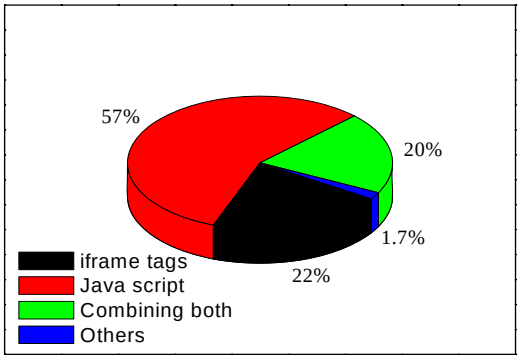

Fig. 2. Distribution of abnormal visibilities

The results of the detection for the proposed system and the Sucop are shown in the Fig.3 below.

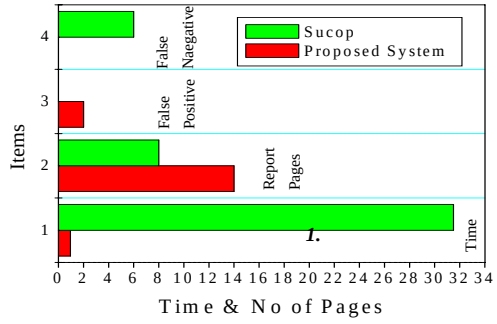

Fig. 3. Performance results of the systems

It is inferred from the figure3 that the performance of the proposed system is higher compared to Sucop. By including the honey client in the detection method, the speed of large number of malicious website detection can also be increased in shorter time duration along with this high performance.

Accuracy rate of the malicious website detection can be evaluated by using the following confusion matrix.

Table 1. Confusion Matrix

		Prediction	
		Malicious Website	**Normal Website**
Actual	**Malicious Website**	True Positive (TP)	False Negative (FN)
	Normal Website	False Positive (FP)	True Negative (TN)

Hence accuracy rate can be evaluated as follows.

$$\text{Accuracy rate} = \frac{TP + TN}{TP + TN + FP + FN}$$

CONCLUSION

After analyzing and viewing the statistics of web malicious codes, an integrated approach of using honey client with an abnormal visibility detection method is proposed to detect the malicious websites effectively and efficiently. A system is developed and implemented based on the integrated approach. It shows that the system can detect the malicious websites in an increased rate and speed along with the high performance and less maintenance cost. It can detect almost all the Web pages containing malicious codes and its location. The system can also be used to track the security state of the target websites and alarm the end users before visiting the malicious Web pages.

Accuracy rate of the malicious website detection can be evaluated using the confusion matrix.The Malicious Display Style can be further improved by extending the abnormal visibilty fingerprints. This can be done by adding new classes that can detect the malicious codes in almost all the tags that holds important data.

REFERENCES

1. ***Ikinci. "Monkey-Spider: Detecting Malicious Web Sites". Sicherheit 2008.***
2. ***Moshchuk, T. Bragin, S. D. Gribble, H. M. Levy. "A Crawler-based Study of Spyware on the Web". In Proceedings of the 13th Annual Network and Distributed Systems Security Symposium (NDSS 2006), San Diego, CA, February 2006.***
3. ***Bin Liang, Jianjun Huang, Fang Liu, Dawei Wang, Daxiang Dong, Zhaohui Liang "Malicious Web Pages Detection Based On Abnormal Visibility Recognition "2009 IEEE.***

4. *C. Seifert, I. Welch, P. Komisarczuk. "HoneyC: The Low-Interaction Client Honeypot". In Proceedings of the 2007 NZCSRCS. Waikato University, Hamilton, New Zealand. April 2007.*
5. *Christian Seifert, Ian Welch, Peter Komisarczuk, "Identification Malicious Web Pages with Static Heuristics ", ATNAC 2008.*
6. *Jianwei Zhuge, Thorsten Holz, Chengyu Song, Jinpeng Guo, Xinhui Han, and Wei Zou "Studying Malicious Websites and the Underground Economy on the Chinese Web ", 2008.*
7. *M. Almgren, H. Debar, M. Dacier, "A Lightweight Tool for Detecting Web Server Attacks." In Proceedings of the ISOC Symposium on Network and Distributed Systems Security, an Diego, CA, February 2000.*
8. *M. Benedikt, J. Freire, P. Godefroid, " VeriWeb: Automatically Testing Dynamic Web Sites". In Proceedings of 11th International World Wide Web Conference (WWW'2002), Honolulu, HI, May 2002.*
9. *Y. Wang, D. G Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, S. T. King, "Automated Web Patrol with Strider Honey Monkeys: Finding Web Sites That Exploit Browser Vulnerabilities". In Proceedings of the Network and Distributed System Security Symposium, NDSS 2006, San Diego, California, USA.*
10. *http://www.StopBadWare.org.*
11. *Sucop. http://www.sucop.com/*
12. *HTMLParser. http://htmlparser.sourceforge.net/.*