Poster: Identify and Track Web Attacker Based on Deceptive Technology and Browser Fingerprint

Heyang Lv[1] Binxing Fang[2] Xiang Cui[3,4]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
[2]Institute of Electronic and Information Engineering in Dongguan UESTC, Guangdong China
[3]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[4]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
heyanglv@126.com fangbx@bupt.edu.cn cuixiang@iie.ac.cn

***Abstract*—At present, the protection of website faces many challenges. First, it is difficult to identify the attacker, because the attack traffic is submerged in a large number of normal traffic. Despite the WAF (Web Application Firewall) is powerful, but exists false positive and may be bypassed due to their own vulnerability. Second, even if the attacker is identified and then blocked by adding the IP address to the blacklist. But next, when they switch to another IP address, how to re-identify the attacker immediately before attacking again? This requires the system has the ability to track the attacker. Therefore, how to identify the attack traffic efficiently and track the attacker are essential to website protection. In this paper, we propose a system model, which combines deceptive technology and browser fingerprint to identify and track web attacker. And it can be as an effective complement to the existing defense mechanism.**

Keywords—website protection; deceptive technology; browser fingerprint;

I. INTRODUCTION

The prototype system works based on the premise or scene that the attacker uses a browser to visit the website at some point in the life cycle of web attack, not just use command line tools, due to the reason that the fingerprint collection script needs to be parsed by the browser.

In the information collection phase, the attacker usually visits the website in order to find possible vulnerability or sensitive information, especially targeted attack. Therefore, based on the idea of deception, the defender can arrange some false information. For example, deploy some forged web pages, which are not visible to the normal user and inserted JavaScript code to collect the fingerprint information of the visitor. Because the forged web page is not visible to the normal user, once visited, the visitor can be considered to have malicious intentions. Then it can block the visitor and store the client fingerprint into library for later use. Next, when the same attacker, who visits the website again with different IP (or visits another website that shares fingerprint information with current website), is considered as an anonymous user now. Then by collecting its fingerprint and comparing the fingerprint with the acquired fingerprints can effectively identify whether the anonymous visitor is an attacker. Because the fingerprints stored in the library are collected from forged web pages, and considered to be associated with an attacker.

The advantage of combining the deceptive technology [1] and browser fingerprint [2] has two aspects. First, the deceptive technology can reduce network noise. Because the false information only can be found through some professional and unconventional means, which is what an attacker might do. Therefore, the use of deception techniques can effectively distinguish the legitimate visitor and the illegal visitor. Second, using JavaScript code to obtain these client attributes which are related to browser fingerprint is the normal function of the browser. These attributes can't be shielded and it is difficult to change them. So use fingerprint to track the attacker is better dependable.

II. SYETEM ARCHITECTURE

A. Overall Architecture

The architecture of the system model is shown as Fig. 1. The web pages on the website server are divided into two categories, one is the forged page, the other is the normal page. All web pages will be inserted the fingerprint collection script, which is a section of JavaScript code. But only the fingerprint from forged page will be stored in fingerprint library. That is to say, the fingerprints in the library belong to the abnormal user.

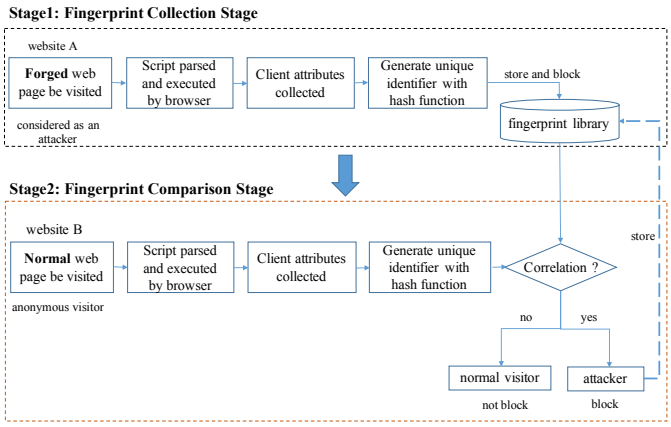

Fig. 1. The architecture of proposed system

And we divide the system running process into two stages, one is called the fingerprint collection stage, the other is called the fingerprint comparison stage. In order to better explain the running process, assume that the two stages occur on website A and website B, respectively. The fingerprint collection stage

refers to collect the client attributes from the user who visits the forged page, generate fingerprint, and then store the data into the library. The fingerprint comparison stage refers to compare the fingerprint information (which is from an anonymous user) with the obtained fingerprints (which are stored in library), calculate their correlation to determine whether the anonymous user is a potential attacker. If associated, then block the user's visit and store the new fingerprint into the library. On the contrary, do not block the user's visit.

III. Implementation

A. Detection of Web Attack using Deception Technique

All deception techniques are based on the idea of "bait", and the following is a list of commonly used deceptive means.

1) Forged subdomain

In the information gathering phase of web attack, collect the subdomain of the target website is an important step. The attacker usually uses subdomain brute force tool. To this situation, the defender can register forged subdomain and bind it to an idle IP address. Then build a website on the server corresponding to the IP and set "*User-agent: * Disallow: /*" in the robots.txt file. When someone detects these subdomains and visits the website, it will be considered as a potential attacker and the fingerprint will be collected and stored.

2) Forged Sensitive Directory

Also during the vulnerability scanning phase of the web attack, the attacker may try to find the sensitive paths of the website by using web directory scanning tool. The sensitive paths may be the backend login address or the path that exists vulnerability. To this situation, the defender could deploy some forged web pages on the website server. And it is best to put the paths of these pages in the robots.txt, because it's the place where an attacker usually checks for collecting information. Once the forged web pages are visited, the visitor is considered as a potential attacker and the fingerprint will be collected and stored.

B. Fingerprint Collection Stage

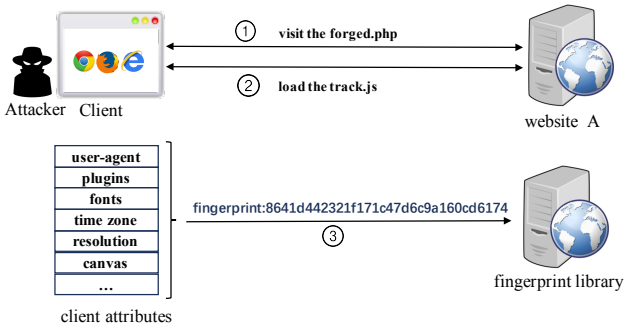

Fig. 2. Fingerprint collection

Through the JavaScript code, it can get the client's user-agent, plugins, fonts, time zone, screen resolution, language, cpu class, etc. And all client attributes will be used to generate fingerprint. The process of the fingerprint collection stage is shown as Fig. 2. First, it supposes that the attacker has been successfully deceived and lured to visit the forged page [①]. Then, the browser parses the response returned by the server and loads the track.js [②]. Last, after the script is parsed and executed by the browser, it will collect the client attributes and a unique identifier will be generated with hash function. Then all data will be post to the backend server and stored in fingerprint library[③].

C. Fingerprint Comparison Stage

As Fig. 1 stage 2, an anonymous visitor visits the website B, which has a cooperative relationship with website A and share the same fingerprint library. So how to confirm whether the anonymous visitor is a potential attacker. It first collects the visitor's client attributes and generate fingerprint, then calculate the correlation of this fingerprint with any record in the fingerprint library. If it matches any record in the fingerprint library, it means that the anonymous visitor is a potential attacker. On the contrary, if not match, it means that the anonymous visitor is a normal user.

When calculate the fingerprint of the client, it first assigns different information entropy to every client attribute and then use hash function to generate the fingerprint. But it exists the situation that two fingerprints are associated and belong to a signal attacker, but not exactly the same. The reason may be that the attacker updates the browser plug-ins, font library or other reason that leading the change of some client attributes. For this situation, to associate different fingerprints from a single client, the similarity measure algorithm can be used to reduce the rate of missing report.

IV. Conclusion And Future Work

In this paper, we present a system model based on deceptive technology and browser fingerprint technology to protect the website. Based on deceptive technology, it can effectively identity whether the visitor is an attacker in a large number of network traffic. And based on browser fingerprint, it can effectively perceive, track and block the attacker no matter how many springboards are used. In the future work, we plan to design more available deception techniques and improve the accuracy of the fingerprint correlation algorithm.

Acknowledgment

This work is supported by the Industry-University-Research Cooperation Project of Guangdong Province "Academician Workstation of Healthcare Cloud Security in Guangdong Province" (No.2016B090921001) and the Ministry of Science and Technology of China (No.2016QY08D1602).

References

[1] Virvilis N, Vanautgaerden B, Serrano O S. Changing the game: The art of deceiving sophisticated attackers[C]//Cyber Conflict (CyCon 2014), 2014 6th International Conference On. IEEE, 2014: 87-97.

[2] Eckersley P. How unique is your web browser?[C]//International Symposium on Privacy Enhancing Technologies Symposium. Springer Berlin Heidelberg, 2010: 1-18.