Lexical Analysis for the Webshell Attacks

Lawrence Y. Deng
The Dept. of CSIE of St. John's University
New Taipei City, Taiwan
lawrence@mail.sju.edu.tw

Dong Liang Lee
The Dept. of IM, St. John's University
New Taipei City, Taiwan
lianglee@mail.sju.edu.tw

Lim Xiang Yann,
The Dept. of CSIE of St. John's University
New Taipei City, Taiwan

Yung-Hui Chen
The Dept. of NCE of Lunghwa University
New Taipei City, Taiwan
cyh@mail.lhu.edu.tw

***Abstract*—In this paper, we proposed a design considering for web application firewall system. Anyway, the website managing, web technology and hacker technology are vastly different from each other, whilst hacker techs are keep on moving forward, while company and government MIS are difficult of updating their information to the latest state due to working, they are often not updated and followed to the latest stuffs. And moreover, these newest techs are in generally needed to be figured out whether it matches the compatibility of the old versions, such as the recent PHP 7, many MIS are not capable to update their software to handle the variants of WAF bypass methods.**

Keywords- Webshell, WAF, SQL Injection, Command Injection, Cross-Site Scripting

I. INTRODUCTION

Due to the growing up of the ubiquitous accessing to the internet, the concept of network and internet are accepted by everyone. That paperwork, data filings were in the past, with more and more companies, industries and government agency stores information in electronic state.

The ascending of the electronic information is inevitable, thus making the safety more and more obvious to be in dire need of protection.

Although there are tons of organization and government agencies that are trying to spread the importance of information security, NGOs such as TDOH, RAT, The Honeynet Project Taiwan Chapter, HITCON, TWCSA, and government organizations such as NCHC,III etc., information security incidents just seem to keep emerging, endlessly.

Taiwan's vulnerability report sites such as VulReport [1] and HITCON Zeroday [2] and even Mainland's Wooyun [3] will show that lots of Taiwan sites are being exploited and posted up by white hats or grey hats daily. E.g.: The recent incident about Taiwan Household Registration Office's site that revealed that data can be inserted arbitrary and many "footprints" (indications of past intruders) are found.

Thus, for web administrators or profession-related police officers to solve cases or in the need of investigation, they need to spend a lot of time to analyse which sites are injected with malicious codes (hobbyhorse backdoor, a.k.a. one-sentence backdoor, which is a trend for injecting due to the nature of it only contains one sentence, it is easy to inject) [10], but they aren't all familiar with cyber-attacks, so we wish to assist them on finding which page contains those codes. Webshell Illustration was showed in figure 1.

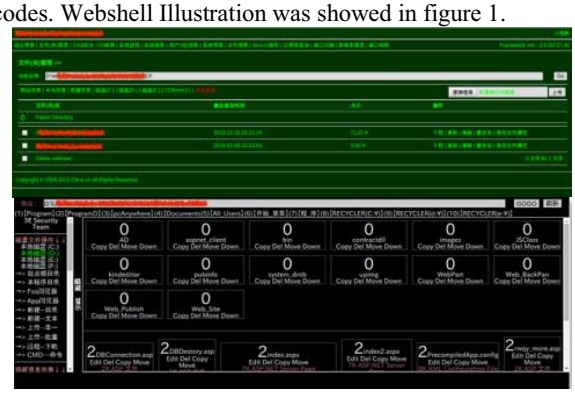

Figure1 Example of Webshell

II. PROTECTION SOFTWARE V.S. WAF

Currently, there are lots of antiviruses available, such as AVAST!, Norton, Kapersky, 360 etc., but for the defence and monitoring on webshells are relatively few compared. Some recognized amongst the few are as known as Web Application Firewall, WAF, e.g., CloudFlare [4], Amazon WAF/AWS WAF [5], SafeDog, 360WebGuard and KnownSec. The main goals of the Web Application Firewall (WAF) [7, 9, 12, 13, 14] is to defence the attack by the web application e.g. cross-site scripting and SQL injection [8, 11, 15].

The following attacks might injure our system:

1). Cross Site Scripting: or named the XSS attack, to re-direct the user to the forgery site and to the username, password and then to get benefit of what they wanted. It is one of the famous attack methods.

2). Encoding: encoding the target system instructions to avoid the system security check or make the system program failed.

3). Header Tampering: tampering the request header to cheat the security system.

978-1-5090-3071-2/16 $31.00 © 2016 IEEE
DOI 10.1109/IS3C.2016.149

4). Path Traversal: traversal the active directory and attempt to access the secret files.
5). SQL Injection: one of the most famous attacks methods to access the database or get the root authorization and even take down the whole system.
6). Remote Command execution: attempt the website's leak and to execute the command to injure the system.
Probes: search the weakness of the system and might inform the others hackers to attack the system.
7). Known Worms: Worms for denial of service or to transfer huge nonsense information to influence the system normal function.
8). Compromised Servers: a bad management system even didn't reset the password that was easy to attack and to get the authorization.
9). Denial of Service: This attack is an attempt to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. A distributed denial-of-service (DDoS) is where the attack source is more than one–and often thousands of-unique IP addresses.
10). Session Hijacking: sometimes also called a session key—to gain unauthorized access to information or services in a computer system. In particular, it is used to refer to the theft of a magic cookie used to authenticate a user to a remote server. It has particular relevance to web developers, as the HTTP cookies used to maintain a session on many web sites can be easily stolen by an attacker using an intermediary computer or with access to the saved cookies on the victim's computer.
11). Cookie Tampering: The user browser is supposed to send the cookie back to the web site (or another web site in the same domain) unaltered with its subsequent page requests. Cookies are used by web sites for authentication, session management, tracking, personalization and so on. All these uses rely on the presupposition that cookie values are not modified outside web sites.

dotDefender is an enterprise-class Web application security solution that provides Apache and IIS Server Security across Dedicated, VPS and Cloud environments[6].

It prevents Cross Site Scripting (XSS) Attacks, SQL Injection Attacks, Credit Card Disclosure, Denial of Service (DoS) Attacks and more. It meets PCI Compliance and also provides E-Commerce Security, IIS and Apache Security, Cloud Security and more.

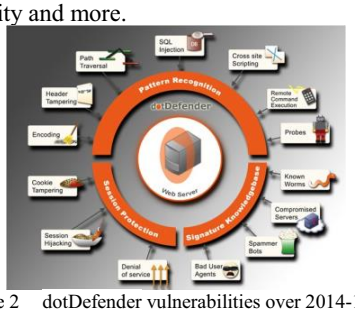

Figure 2 dotDefender vulnerabilities over 2014-15 [6]

The network firewall address in the management of the ports connection (network layer), but the WAF would monitor between the browser (requests) and the web server. The WAF is address in the analysis of the content in the application layer. The feature of the WAF is also different from the Intrusion Prevention Systems, IPS. The IPS is signature-based to compare the dangerous attacks, but the WAF is aim to analysis the behaviors that could protect your own innocence of the web-site's leak. The basement works to exam the input messages were the important first step. The WAF system should analysis the lexical the input messages were illegal or not. The programmer's work included all the examining tasks. The hacker used some weakness of the examining tasks and then to attack the website or database. Therefore, the WAF provided the positive security model to do the input validation. Only the legal messages could pass. This principle called the white-listing that is the common security rule for the application layer. The whiting-listing should consume more system efficiency. Therefore, positive security model might consider the signatured-based subsystem to decrease the computation load for the WAF system. This toward improving the WAF/IDS throughput, they either require significant modifications to existing services by including new content matching algorithms, or require dedicated hardware resources to achieve acceleration, or need to coordinate tasks and aggregate output from parallel processing units [17]. Kai Hwang proposed the hybrid system combines the advantages of low false-positive rate of signature-based intrusion detection system (IDS) and the ability of anomaly detection system (ADS) to detect novel unknown attacks [16].

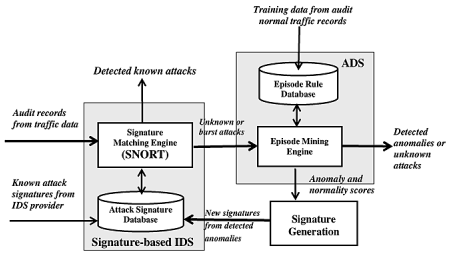

Figure 3 A weighted signature generation scheme. [16]

Wei Xu et al [18] have provided a lexical confinement policy to analysis SQL and command injection Attacks. This policy needs that tainted data should not span multiple tokens. In Figure 4, note that in the benign case, tainted data is confined to a single token that corresponds to a shell parameter. However, in the attack case, tainted data overflows beyond this token — in fact, it creates four additional tokens. A more general criterion was proposed by Su et al [19], which was called syntactic confinement. The figure 4 shown examples of a benign shellcommand (left) and a command involved in an injection attack [7].

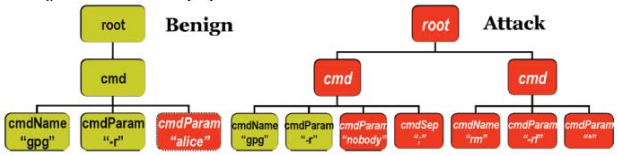

Figure 4 Examples of a benign shellcommand (left) and a command involved in an injection attack [7].

The most popular and widely used among in Mainland China are SafeDog and 360WebGuard, and they cooperate with many industries, including the renowned AliCloud, Tencent and Huawei.

This is a screenshot of an attempt that was intercepted by SafeDog when trying to upload a webshell via eWebEditor (as shown in figure 5).

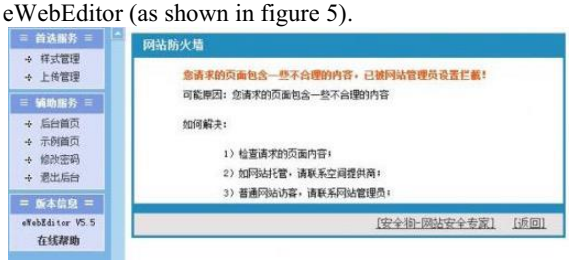

Figure 5 This is a screenshot of an extension list that SafeDog filters

Generally the solutions that SafeDog uses are disabling specific extensions and attributions checking. The figure 6 illustrated the SafeDog's filters.

Figure 6 Illustrated the SafeDog's filters (for specific extensions and attributions checking).

This method is capable to wind down about 60% of the attacks that involves script kiddies, newbies and unskilled white hats or black hats.

III. BYPASS WEBSHELL ISSUES

To hackers, SafeDog's most hated part is that if one attempts to upload a webshell, it is blocked, or if a webshell is successfully uploaded, it can't execute or function properly.

Example, using hobbyhorse to bypass SafeDog

But, is it enough that by using those softwares (automated scanning/auditing software), those webshells and hobbyhorses and their variants are nowhere to run?

Not by cutting into the answer, let's see some examples, and here we should take in SafeDog. A little research would eventually lead to lots of ways for bypassing SafeDog

Example: Due to some incapabilities in handling special characters by using some quite unsafe method, SafeDog filter fails to function properly (for IIS) [19].

A vuln report of SafeDog ignoring file size larger than 1024*1024 bytes. Or such as in this incident (http://tieba.baidu.com/p/3398156228), SafeDog blocks the upload of a webshell. But this time the hacker writes the webshell into a file called 1.txt, and by using include method to let the 1.txt make its way to the include folder(and viola!)

IV. LEXICAL ANALYSIS

To improve this situation, we dug in to a project that was published by tennc (https://github.com/tennc/webshell "38 commits to master since this release"), authors gathered common site webshells and webshells that are manipulated based on hackers, and realizing although most webshell don't share the same signature to each other, but the commmon side is that their base function are typically the same, when they need to access some system level commands, and this is what most webshells basic functionality starts.

Static Code Analysis (also known as Source Code Analysis) is usually performed as part of a Code Review (also known as white-box testing) and is carried out at the Implementation phase of a Security Development Lifecycle (SDL). Static Code Analysis commonly refers to the running of Static Code Analysis tools that attempt to highlight possible vulnerabilities within 'static' (non-running) source code by using techniques such as Taint Analysis and Data Flow Analysis.

Although webshell can run all sorts of system commands, it's still a web-based page. To control a system's shell, the page need to execute some system call functions, such as exec, passthru, shell_exec, system, proc_open, set_time_limit, base64_decode, eval, and by using function to determine whether it is a webshell

Lexical Analysis converts source code syntax into 'tokens' of information in an attempt to abstract the source code and make it easier to manipulate.

Pre tokenised PHP source code:

<?php $name = "Ryan"; ?>

Post tokenised PHP source code:

T_OPEN_TAG

T_VARIABLE

=T_CONSTANT_ENCAPSED_STRING;

T_CLOSE_TAG

Scales Well (Can be run on lots of software, and can be repeatedly (like in nightly builds))

For things that such tools can automatically find with high confidence, such as buffer overflows, SQL Injection Flaws, etc. they are great.

Anyway, many types of security vulnerabilities are very difficult to find automatically, such as authentication problems, access control issues, insecure use of cryptography, etc. The current state of the art only allows such tools to automatically find a relatively small percentage of application security flaws. Tools of this type are getting better, however. That may take the high numbers of false positives. Frequently can't find configuration issues, since they are not represented in the code. Difficult to 'prove' that an identified security issue is an actual vulnerability.

Many of these tools have difficulty analyzing code that can't be compiled. Analysts frequently can't compile code because they don't have the right libraries, all the compilation instructions, all the code, etc.

Moreover, it is obvious that if a page contains lots of "flower codes" (codes that are unreadable or hard to read), encrypted codes and special characters, that page is probably a webshell Just as https://gihub.com/tennc/webshell/blob/master/php/BNKQbAKQ.txt states that there are lots of encrypted texts.

Hackers will encrypt their webshell contents if they found out that there are WAF/related software implemented.

Although it is quite impractical to say that files call system functions are webshells, but through Deletion method (In normal websites, there is not common to use system call functions, or if when a site has just launched, generate the site's MD5, while it changes or some web content changes, indicates that there is page(s) added, modified or deleted, and to list them all out), web administrators have a high success chance of determining whether a webshell.

Some page may need require those function in the forehand(though they are not commonly applied in practical, for security reasons and rarely used), at this time only that they need to filter out the relevance of the site, webshell will be easy to identify. As pages are normally connected to each other for users' convenience, by combining access logs, sorting out view counts and removing static files, filtering out system call files which are relatively getting fewer views than others, and what's left have a higher chance of being a webshell.

V. CONCLUSION AND FUTURE WORK

Website managing, web technology and hacker technology are vastly different from each other, whilst hacker techs are keep on moving forward, while company and government MIS aren't capable of updating their information to the latest state due to working, they are often not updated and followed to the latest stuffs. And moreover, these newest techs are in generally needed to be figured out whether it matches the compatibility of the old versions, such as the recent PHP 7, many MIS are not capable to update their software to handle the variants of WAF bypass methods.

Lexical analysis can't solve all our security problems. thestatic analysis tools look for a fixed set of patterns, or rules, in the code. Although more advanced tools allow new rules to be added over time, if a rule hasn't been written yet to find a particular problem, the tool will never find that problem.

No matter how Webshell changes, to system it is just a file, it still need HTTP servers such as Apache or IIS to compile and turn into a web page, and by this we only need to check to files content, most of the Webshells will be eliminated. At this time, getting access to newest information by public platform, as example github's git clone, they can get the newest methods of payloads and pre-test for their own sites, which can also make them follow up the latest information and expanding their knowledge. By combining drawing and managing a control panel, it will be quite user-friendly to MIS, related investigation departments and even to normal users.

REFERENCE:

[1] https://vulreport.net

[2] https://zeroday.hitcon.org

[3] http://www.wooyun.org

[4] https://www.cloudflare.com

[5] https://aws.amazon.com/tw/waf/

[6] https://www.applicure.com/about_dotdefender

[7] R. Sekar, "An Efficient Black-box Technique for DefeatingWeb Application Attacks", 16th Annual Network & Distributed System Security Symposium Proceedings 2009, USA.

[8] Gregory Buehrer, BruceW.Weide, and Paolo A. G. Sivilotti. "Using parse tree validation to prevent sql injection attacks", In SEM '05: Proceedings of the 5th international workshop on Software engineering and middleware, pages 106–113, New York, NY, USA, 2005. ACM.

[9] Zhendong Su and Gary Wassermann. "The essence of command injection attacks in web applications", In POPL '06: Conference record of the 33rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages, pages 372–382, New York, NY, USA, 2006. ACM.

[10] Wei Xu, Sandeep Bhatkar, and R. Sekar. "Taint-enhanced policy enforcement: A practical approach to defeat a wide range of attacks", In USENIX Security Symposium, August 2006.

[11] P. Madhusudan Sruthi Bandhakavi, Prithvi Bisht and V.N. Venkatakrishnan. "Candid: Preventing sql injection attacks using dynamic candidate evaluations", In CCS, 2007.

[12] E. Kazanavicius, V. Kazanavicius, A. Venckauskas, R. Paskevicius, "Securing Web Application by Embedded Firewall", The research journal ELEKTRONIKA IR ELEKTROTECHNIKA, 2012.

[13] Desmet l., Piessens F., Joosen W., Verbaeten P. Bridging, "the gap between web application firewalls and web applications // Proceedings of the fourth ACM workshop on Formal methods in security. – Alexandria, Virginia, USA, 2006. – P. 67–77.

[14] Krueger T. , Gehl Ch., Rieck K., Laskov P. TokDoc: a self–healing web application firewall // Proceedings of the 2010 ACM Symposium on Applied Computing. – Sierre, Switzerland, 2010.

[15] Moosa A. Artificial Neural Network based Web Application Firewall for SQL Injection // World Academy of Science, Engineering & Technology, Vol. 64. – P. 12–21. 2010.

[16] Kai Hwang, Min Cai, Ying Chen, Min Qin, "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes", IEEE Transactions on Dependable and Secure Computing 2007 vol.4 Issue No.01 - January-March.

[17] Mueen Uddin, Kamran Khowaja and Azizah Abdul Rehman, "Dynamic Multi-Layer Signature Based Intrusion Detection System Using Mobile Agents", International Journal of Network Security & Its Applications (IJNSA), Vol.2, No.4, October 2010.

[18] http://www.360.com/

[19] http://www.secpulse.com/archives/39570.html

[20] http://www.wooyun.org/bugs/wooyun-2015-0133898